Distributed Information Processing

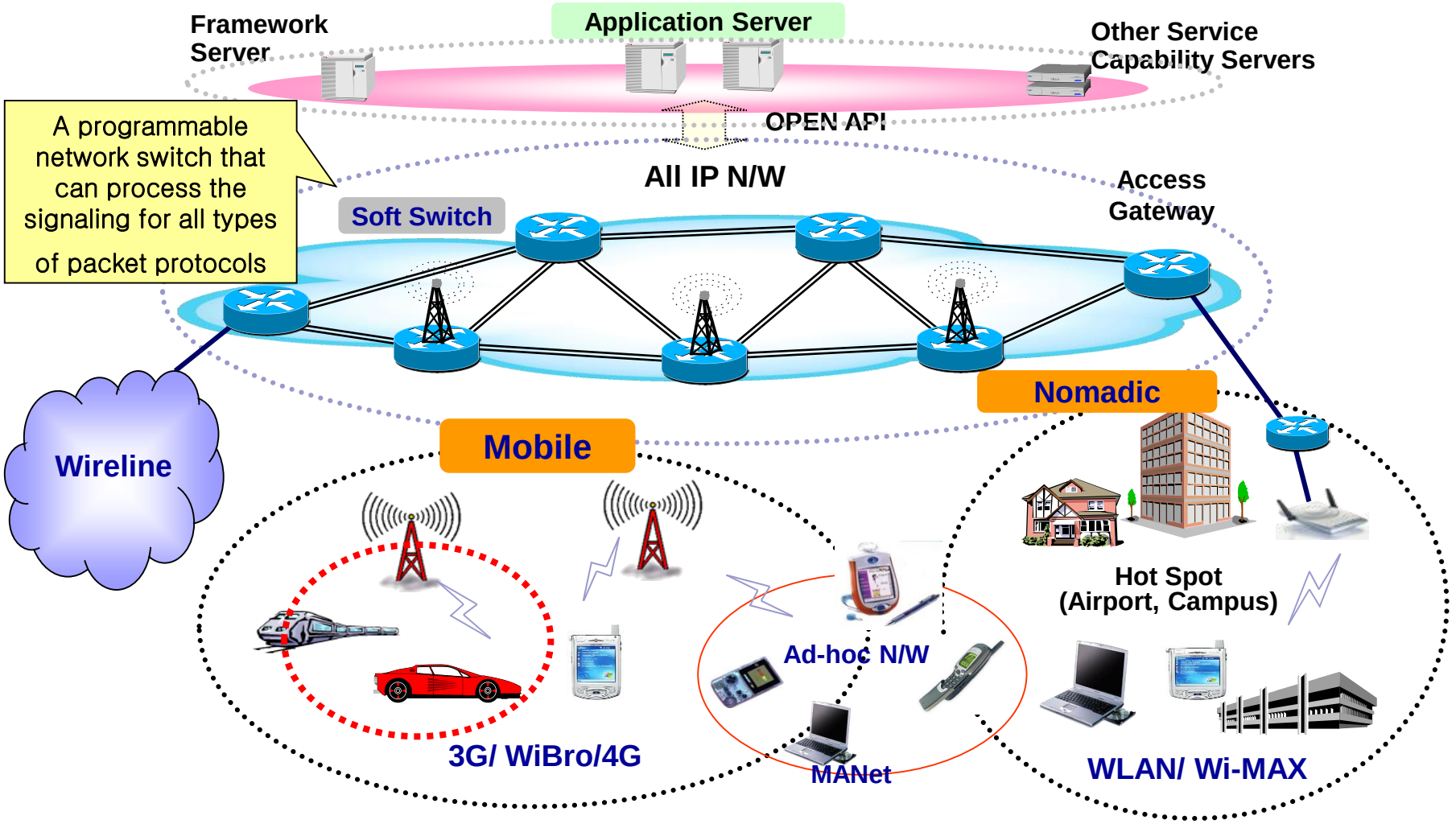
22nd Lecture

Eom, Hyeonsang (엄현상)
Department of Computer Science & Engineering
Seoul National University

Outline

- Information Transmission and Use
  - Convergent Networks
  - Introduction to the Semantic Web
  - A Portrait of the Semantic Web in Action
- Performance Evaluation
- Dynamic Adaptation
- Q&A

Convergent Networks



Future of Search Technology [Brewer02]

- Integration of Textual Search and Database Technologies
- Distributed Repositories
- Context
- Integration with the Physical World
- Novel User Interface
  - To Avoid Information Overload
- Personalization
- Bias

Semantic Web Basics [Gruninger02]

- Ontology
  - Formal Explicit Specification of a Shared Conceptualization
    - Conceptualization: how people think about things in a particular subject area
    - Explicit Specification: concepts and relationships of the abstract model given explicit terms and definitions

Semantic Web Basics (Cont'd)

- Ontology Uses

Uses of Ontology (customized from the uses of ontology identified at the KRSL kickoff meeting 1994).
For communication
between implemented computational systems.
between humans.
between humans and implemented computational systems.
For computational inference
for internally representing and manipulating plans and planning information.
for analyzing the internal structures, algorithms, inputs and outputs of implemented systems in theoretical and conceptual terms.
For reuse (and organization) of knowledge
for structuring or organizing libraries or repositories of plans and planning and domain information.

Ontology Applications: Knowledge Management and the Semantic Web

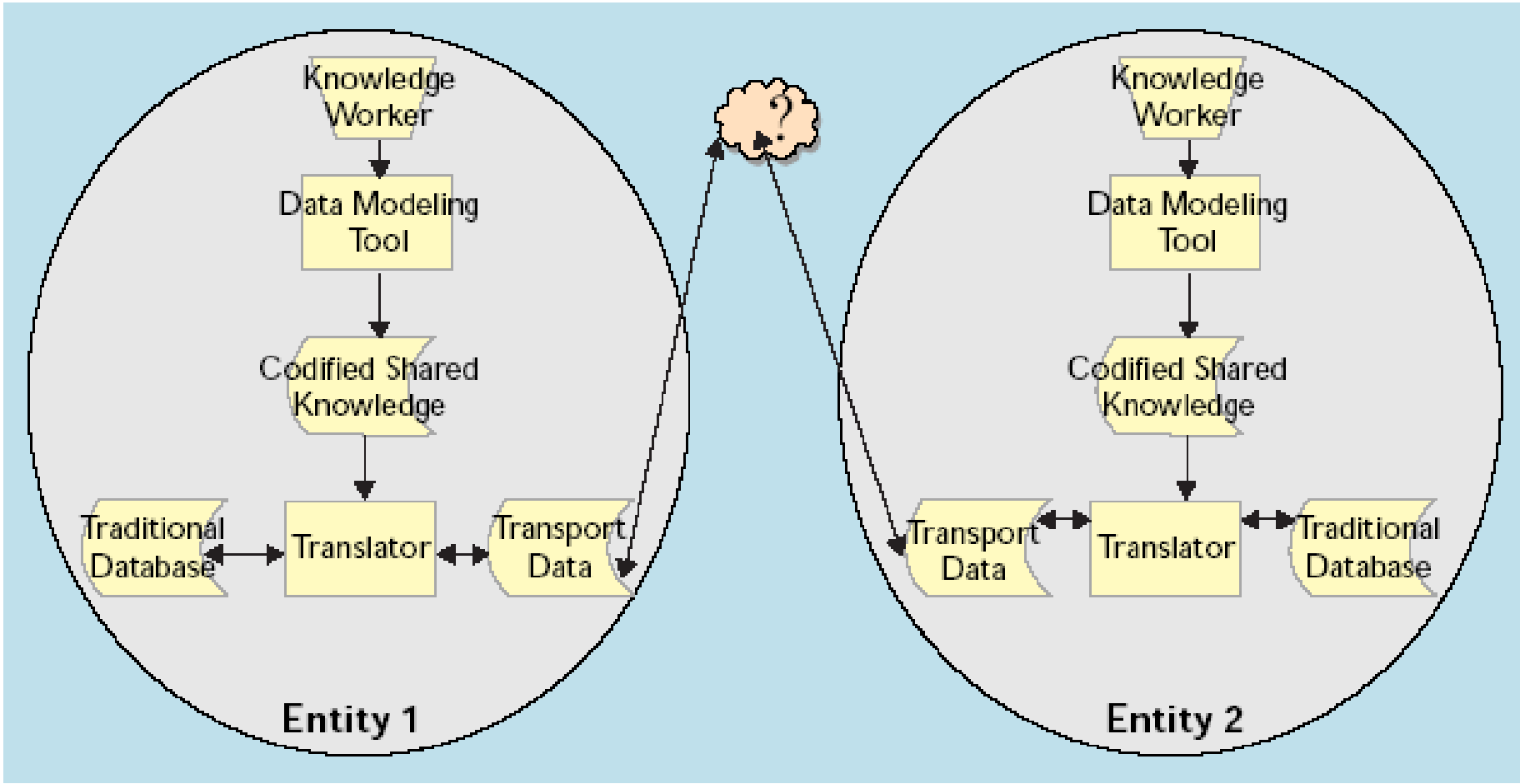
XML vs Ontologies [Kim02]

- Commonality
  - Means of Explicitly Representing Information Applied So That a Reader Interprets Shared Data As Intended by the Data Author
- Differences
  - Need for the Same Understanding
    - XML requires it while ontology does not
      - E.g., <foo>7</foo>
  - Complexity
    - Semantics are not represented with XML use
  - Efficiency vs Interpretability
    - Reducing Complexity vs Reducing Uncertainty

XML Assumes the Same Understanding of What "foo" Means, But "foo" Can Be Defined in Ontology Use

Using Ontologies for Uncertainty Reduction [Kim02]

- Case Where Ontology Is Appropriate



Example CS Department Ontology

Name: cs-dept-ontology
Version: 1.0
Extended Ontology
Base Ontology (base-ontology, version 1.0)
ISA Hierarchy (Taxonomy)

- Person
  - Worker
    - Faculty
      - Professor
    - Assistant
    - AdministrativeStaff
  - Student
- Organization
- Publication
- Schedule

Relationships

Relation	Arg1	Arg2
PublicationAuthor	Publication	Person

Inferences

- Suborganizations are transitive
- Affiliations are invertible
- Membership transfers through suborganizations

For the Semantic Web, an Ontology Must Be Expressed in a Formal Language So That a Given Ontology Expression Can Be Interpreted and Processed Unambiguously by a Machine

Ontology Issues [Kim02]

- Designing an Ontology Development Tool
  - Useful and Usable to a Knowledge Worker
- Developing of Decentralized and Adaptive Ontologies
  - To Be Used in Combination with Other Ontologies
    - Use of Ontologies for Software Specification

Performance Evaluation

- Exploiting Parallelism with the Distributed System (Compiler or Library)
  - Autoparallelization
    - Heterogeneity
    - Variable latencies
  - Manual Computation Decomposition and Load Balancing (Distributed Memory)
    - Architecture independence
  - Data Allocation (Distributed Memory)
    - Maximizing locality
    - Minimizing communication

Data Parallelism

Performance Evaluation (Cont'd)

- Parallel Execution of Components
  - Load matching
  - Communication optimization
- Overlap of Communication with Computation (Distributed Memory)
  - Large and variable latencies
- Reuse of More Data in Local Memories (Distributed Memory)
- Spreading Computation Evenly across Processors (Distributed Memory)
  - Minimizing communication

Task Parallelism

Latency Hiding

Latency Reduction

Load balancing

Performance Evaluation (Cont'd)

- Performance Tools
  - Goal
    - User's identifying and overcoming performance bottlenecks
  - Functionalities
    - Measurement
    - Analysis
    - Visualization
    - Engineering/Tuning
    - Estimation/Prediction

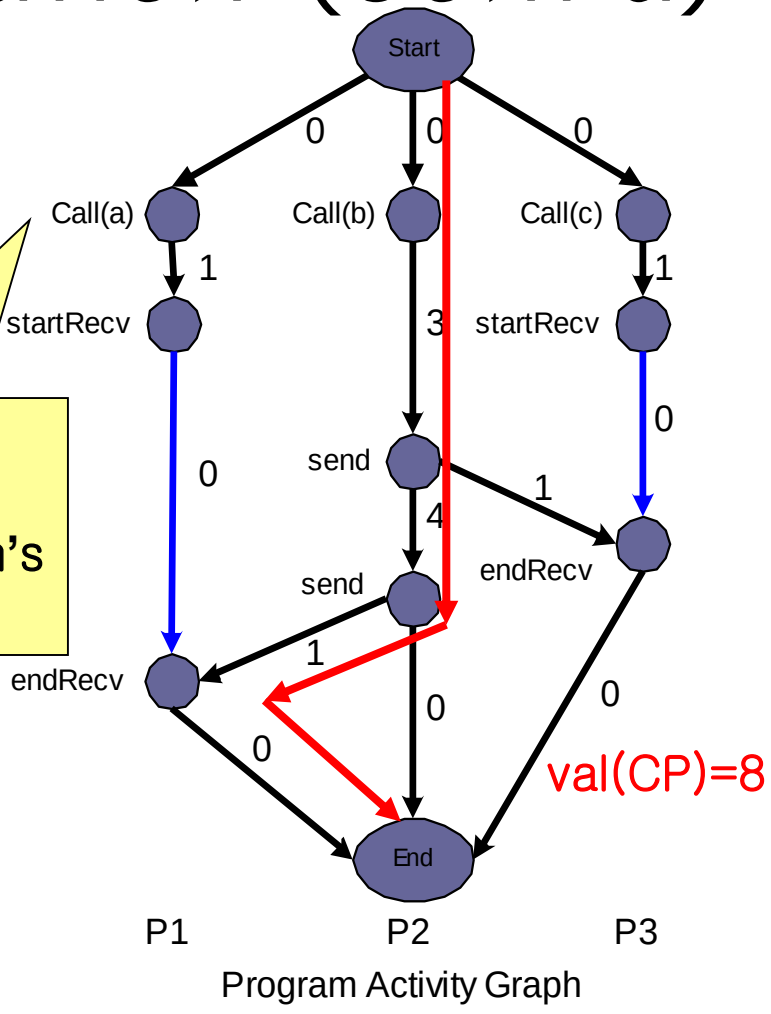
Via Instrumentation

To Identify Bottlenecks

Performance Evaluation (Cont'd)

- Critical Path
  - Longest Path through the DAG
    - Corresponding to the longest path
- Critical Path of a Program
  - Longest CPU or Communication Weighted Path of the PAG

Improving This Procedure May Not Improve the Program's Execution Time



Program Activity Graph

a Graph Consisting of Nodes Representing Significant Events, and Arcs Indicating the Ordering of Events within a Process or Synch Dependencies between Events

Dynamic Adaptation

- Adaptivity
  - Adaptation of Applications to Changes in Their Execution Environment
    - Changes in computational load
    - Changes in network performance
- Application-Specific Framework
  - Working Cooperatively with:
    - Measurement and monitoring
    - Alternative evaluation and selection
    - Performance-driven scheduling

Support at Different Levels

Prior Identification or Provisioning

Resource Reservation

Dynamic Adaptation (Cont'd)

- Changes in Computational Load
  - Obtaining Additional Resources
  - Asking Other Components to Adapt for Improvement
  - Relocating Computation
  - Reducing Requirements in Areas of Little Interest

Introduction (Cont'd)

- Changes in Network Performance
  - Controlling Error
  - Making a Bandwidth Reservation
  - Making Other Links Ask for More Bandwidth
  - Relocating Communication
  - Applying Compression

Dynamic Adaptation (Cont'd)

- Adapting Applications
  - Bandwidth Adaptation Approaches
    - Resource reservation (problematic)
      - Consumption of large memory for storing flowspecs
      - Low utilization for guaranteed services
      - Not being supported by nonswitched Ethernets & wireless LANs
      - Need for deploying policy control, security, and charging mechanisms
    - Adaptation of applications' requirements
  - Bandwidth Adaptation Requirements
    - QoS Measurements
      - E.g., RTCP (Control Protocol) in RTP for continuous media

Dynamic Adaptation (Cont'd)

- Delay Adaptation
  - Goal of Using Large Playout Buffers
    - Conversion of a variable delay into a fixed delay
    - Starvation prevention
  - Support for Using Large Playout Buffers
    - Variable buffer requirement estimation
    - Fixed Buffer
- Loss Adaptation
  - Retransmission (Limited by Delay Tolerance)
  - Redundant Transmission
  - Interleaving
  - Forward Error Correction (for Perfect Reconstruction with Redundant Parity Packets)

Application Adaptation

- Goal
  - Performance Contract
    - Quantified expectations between application performance demands and resource service capabilities

As a Form of Service-Level Agreements

- Techniques
  - Contract Monitoring
    - Verifying, detecting when, and diagnosing why
  - Adaptive Control
    - Adapting to a new resource regime

C.f., Migrating to Other Resources vs Adjusting Contract Parameters Dynamically

Resource and Performance Variability

- Sources of Variability in Performance and Availability
  - Contention
  - No Support for Reservation
  - Failure and Preemption
- Adapting the Execution for High Performance in a Shared Environment
  - Relocating Resources
  - Changing Application Behavior

Variability (Cont'd)

- Problem of "Static" Performance Models
  - Working Only under Ideal Conditions
    - Computational speeds
    - Network latency and bandwidth
    - I/O speed
  - Not Working under Dynamic Conditions
- Adapting Dynamically to Changing Conditions
  - Acquiring New Resources
  - Reducing Solution Resolutions
  - Switching to Alternatives

Instrumentation and Metrics

- Instrumentation
  - Automatic
  - Minimizing Perturbation and Intrusion
  - To Be Inserted at the Proper Level
- Metrics
  - To Be Selected Appropriately
  - Considering Measurement Uncertainty and Temporal Variability
    - Tradeoff between the length of measurement interval and adaptability

E.g., for Contract Verification and Validation

Adaptive Control Example

- Real-Time (Runtime) Monitoring
  - Instrumentation
    - E.g., inserting "sensors"
  - Periodic Transmission of Sensor Data
  - Analysis of the Data
    - E.g., evaluation of the rule base with the data, and detection of contract violation
  - Notification of the Result
    - E.g., distributing the result via the sensors

Issues: e.g., Assessing Temporal Variability and Contract Violation

Adaptive Control Example (Cont'd)

- Remediation
  - Halting the Execution
  - Migrating the Workload
    - At different levels
    - Comparing the benefit & cost
  - Restarting the Application

Requiring the Support Such as Checkpointing

Requiring Control Stability and Rescheduling Mechanisms

References

- [Brewer02] Eric A. Brewer, "The Consumer Side of Search," CACM, Vol. 45, No. 9, September 2002. pp. 41
- [Gruninger02] M. Gruninger and J. Lee, "Ontology Applications and Design," CACM, Vol. 45, No. 2, February 2002. pp. 39-41
- [KIM02] H. Kim, "Predicting How Ontologies for the Semantic Web Will Evolve," CACM, Vol. 45, No. 2, February 2002. pp. 48-54

References (Cont'd)

- [Foster99] I. Foster and C. Kesselman (Editors), *The GRID: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann Publishers, November 1999
- [Foster03] I. Foster and C. Kesselman (Editors), *The GRID 2: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann Publishers, November 2003