Toward Deeply Adaptive Societies of Digital Systems

Antonio Carzaniga
University of Lugano

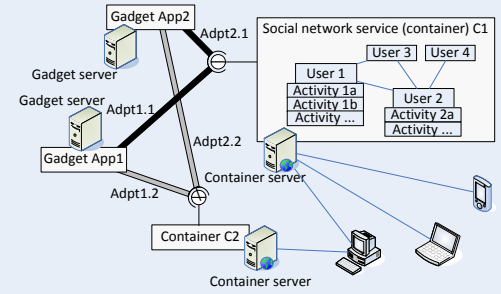
Giovanni Denaro
University of Milano-Bicocca

Jacky Estublier
Lab Logiciels Systemes et Reseaux Grenoble

Mauro Pezzè
University of Lugano, University of Milano-Bicocca

Alexander L. Wolf
Imperial College London

Scenario: An Adaptive Social Network API



Standard API, e.g., the OpenSocial API
Gadget local adaptors, e.g., WS mediator for unsupported activity fields

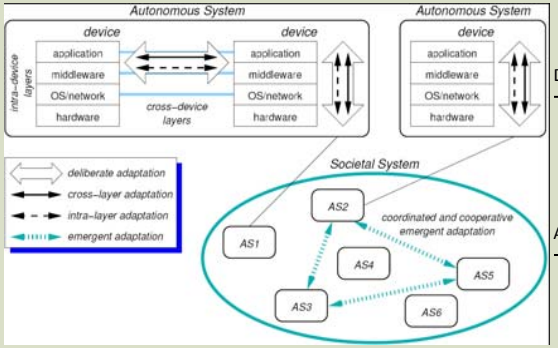
PROVIDED ADAPTATION MECHANISMS
- Detect API-related integration mismatches; correspondingly deploy service-mediators between clients and servers
  Scope: cross-devices, intra-layer (application)
  Goal: guarantee robust interoperability

POSSIBLE CONFLICTS
Mediators that modify server-side data to solve the mismatches of an application, may negatively affect the behavior of other applications

EXAMPLE
Gadget App1 uses the URL activity field. The adaptation mechanism detects a mismatch if this field is not supported by a server; it deploys a service-mediator that stores URLs in a supported field and restores the URL field during retrieving.
However, gadget App2 is using the activity field too. App2 ignores the existence of the service-mediator triggered by App1, thus ending up with retrieving inconsistent data

Societies of Digital Systems

Exploit intra/cross layer/device adaptations to guarantee adequate dependability



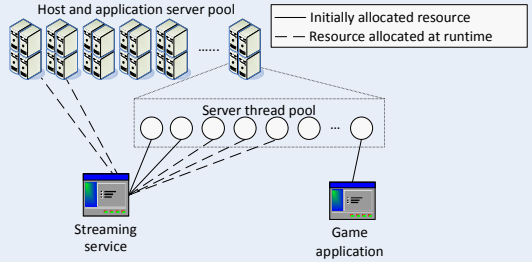
Deliberate design and assembly
→ Administrative control

Ad-hoc/emergent interactions
→ "anarchic" regime

More and more pervasive
Open systems, environments, cooperation
Exploit dynamically discovery and integration of third-party services
Manage/tolerate inter-system incompatibilities
Afford dynamic/unpredictable/fail-prone/reconfigurable environments and scarcely available resource
Exhibit unforeseen global and local behaviors due to side effects and unanticipated interactions

Applications for: domotics, automotive, communication, entertainment, health/medical support environment monitoring, transportation, energy production and management, …

Scenario: An Autonomic Data Center



PROVIDED ADAPTATION MECHANISMS
- The higher the workload, the larger the thread pool for an application
  Scope: intra-device, intra-layer (middleware)
  Goal: optimize performance
- Redundate heavily loaded applications
  Scope: cross-device, intra-layer (middleware)
  Goal: optimize performance

POSSIBLE CONFLICTS
Instances of both above mechanisms for an application, may result in over-reaction

EXAMPLE
An application runs on a host over T threads. When the workload increases, the first adaptation mechanism doubles the threads, while simultaneously the second adaptation mechanism replicates the application on a second host, thus resulting in an unwanted triple resource allocation that may affect performance of other applications in the data center

The coexistence of multiple adaptations may generate conflicts between mechanisms with different scopes or under the control of different components/applications

Universal controllers are not a viable solution

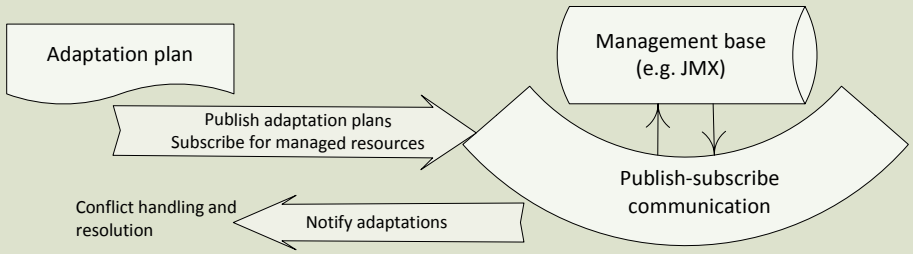
Ad-hoc controllers need coordination to avoid conflicts

DEEP ADAPTABILITY

Ability to adapt both as individuals and collectively, integrating adaptation mechanisms across devices, systems and architectural layers

Our idea: A reference architecture for deep adaptability

1. Exploit mature management technologies to manage system status information
2. Represent plans as change sets over the management base
3. Use content-management publish/subscribe infrastructure to distribute relevant information and allow for conflict resolution



The path ahead (research agenda)

Investigate the suitability of application management technologies
- Ability of storing both adaptation and application state information
- Representing time dimensions of adaptation plans: scheduled time, duration, status (in-progress, committed, …)
- Are ontologies and ontology research players?

Investigate relationships with mature management technologies from other fields
- Network management, operating system management, ...

Investigate distributed conflict resolution strategies
- Local/deterministic handling of conflicts (≡ syntactic local merges) (e.g., all applications but the one with highest identifier abort)
- Context-dependence, negotiation requirements

Enrich evidence: more and more scenarios and case studies