A Human-centric framework for universal access

Canadian Undergraduate Software Engineering Conference March 7-9, 2002

Jacob Slonim

slonim@cs.dal.ca

Dalhousie University, Halifax, Canada

Overview

- Historical perspective
- Future trends
- Accessibility problems
- Human-centric architecture
- Functionality of components
- Example interactions
- Conclusion

History (’70s)

- Centralized paradigm
  - Systems developed by computer scientists for computer scientists
  - Resource limitations caused much of the complexity in systems to be pushed out to the end users
  - Computers used by computing professionals

History (’80s)

- Centralized paradigm -> Client-Server paradigm
  - Driven by reduced cost of computing and the resulting increase in client machine capabilities
  - Middleware introduced as a layer of abstraction to reduce complexity for applications programmers dealing with heterogeneous distributed systems
  - Computers used for business process automation

History (’90s)

- Shift in type of individual accessing systems
  - Personal computers proliferate
  - Widespread Internet access from work and home
  - Roll-out of infrastructure based on reduced hardware and networking costs
  - Complexity of systems slows roll-out for some sectors of society

Today

- Growing disparity between technological “haves” and “have nots” (Digital Divide)
- “New economy” must be more inclusive
- How can we make Electronic Commerce as accessible as the telephone?

Evolutionary approach to development

- Hardware is replaced over time
  - Advances in hardware take replacement approach while maintaining backwards compatibility.
- Software evolves over time
  - Many systems in use today are four decades old
  - Replacement of legacy systems infeasible

Future…

Client-Server -> Peer-to-Peer

- Anticipated shift driven by issues of scalability in client-server systems and by the reduced cost of networking

Technology Push -> Technology Pull

- Allow domain experts to apply technology to problems without the intervention of computing professionals

Computing-Centric -> Human-Centric

- Personalization required to increase accessibility
- Complexity of interaction must match user's capabilities

Personalization

For our purposes, personalization means more than just customizing the look and feel of a web site interface or accessibility techniques such as translating from text to speech

Instead, personalization implies a system focused on the end user's needs, preferences, abilities, and computing context at all times, and the customization of all interactions with the user in a way that reduces complexity for the end user.

Types of Complexity

Interaction complexity – the degree to which the steps to be taken during an interaction are intuitively tailored to the end user

Information overload – too much data relayed to the end user due to insufficient filtering

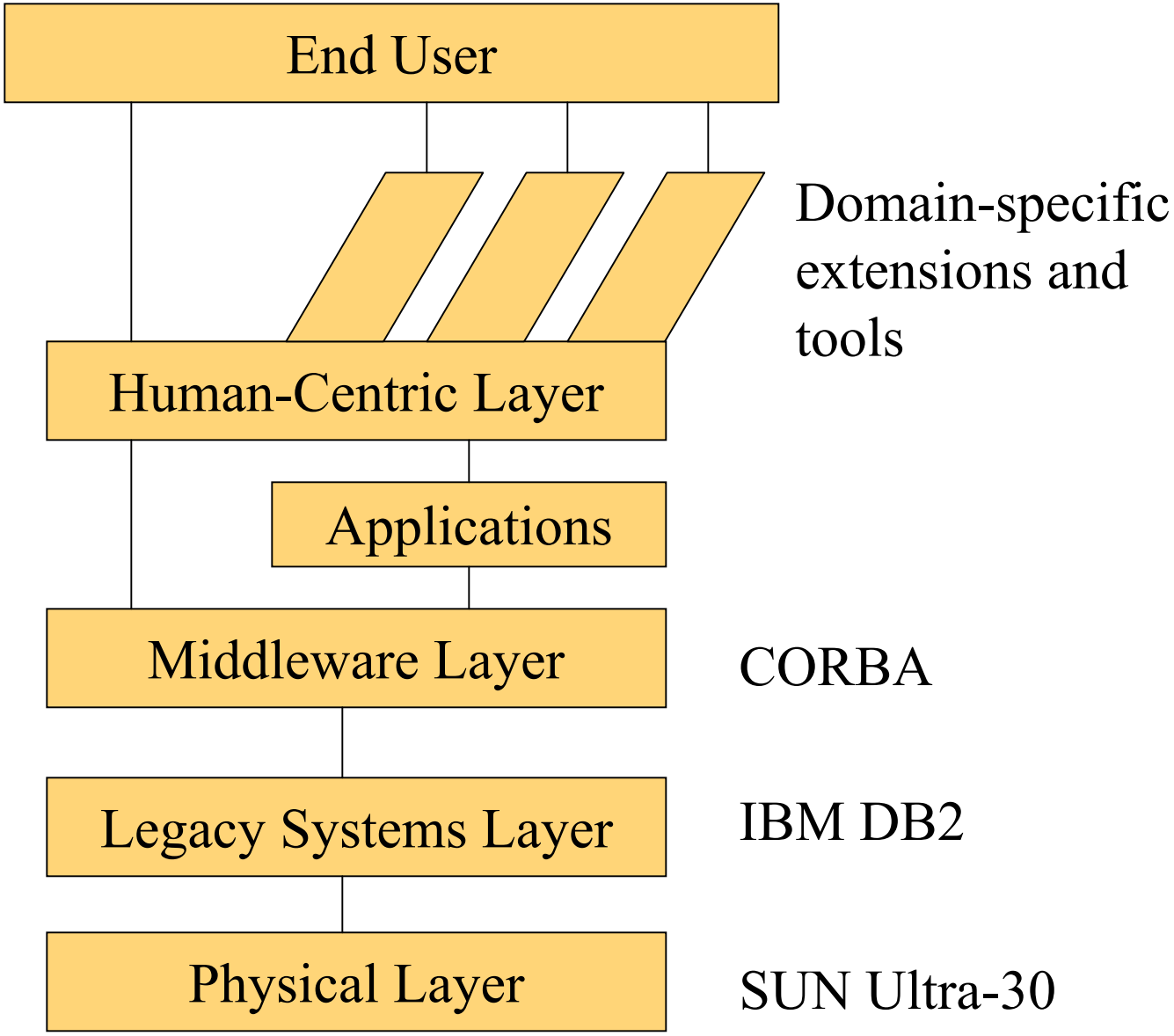
Mental model complexity – the end user's mental model of the system's underlying states can be overly complex

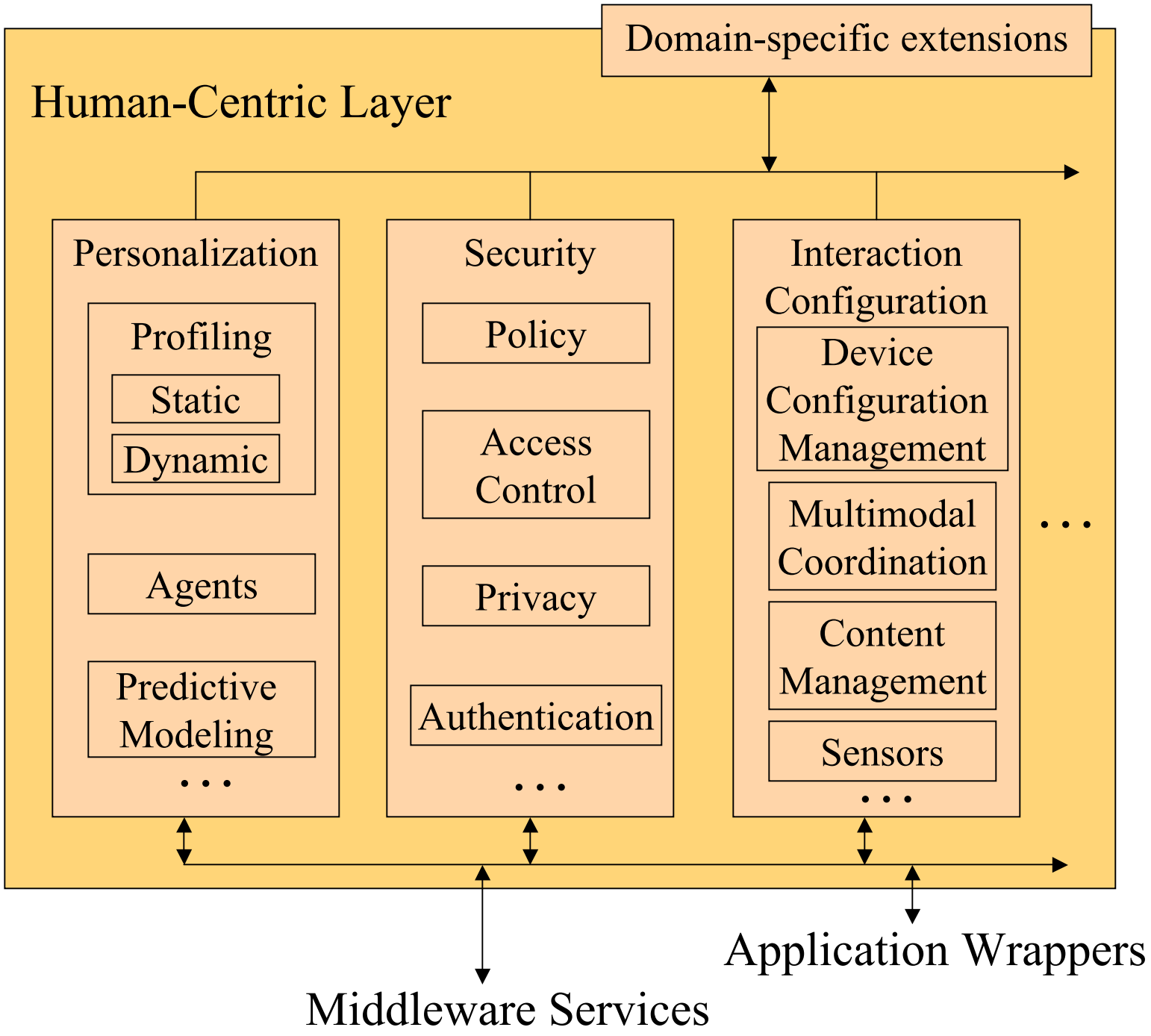
Bridging the Gap

- Need personalized systems to increase accessibility
- Stuck with legacy computing-centric systems
- Introduce a layer of abstraction to bridge the gap

High-level

Architecture




Domain-specific extensions
Human-Centric Layer
Personalization
Profiling
Static
Dynamic
Agents
Predictive Modeling
. . .
Security
Policy
Access Control
Privacy
Authentication
. . .
Interaction Configuration
Device Configuration Management
Multimodal Coordination
Content Management
Sensors
. . .
. . .
Middleware Services
Application Wrappers

Personalization

Profiling

Static

Data that remains constant or changes periodically

Stored in a database in as fine a granularity as possible

Dynamic

Information subject to rapid change, derived from user interactions, datamining, interactions with other modules of the human-centric layer and with domain experts

Agents

Domain-specific agents that operate for the end user

Predictive Modeling

Tools to dynamically learn personalization information

Security

Policy

Rule-based system configurable by end user, with domain-specific management policies provided by domain experts

Access Control

Fine to coarse grained access control mechanisms to restrict and allow access to personalization information

Privacy

Mechanisms to ensure privacy is not violated based on security policy rules (e.g., datamining to protect against query-based attacks on privacy)

Authentication

Configuration of authentication methods for allowing multiple levels of access, including GPS and biometric authentication techniques

Interaction Configuration

Device Configuration Management

Manages internal model of end user operating environment(s)

Multimodal Coordination

Dynamic coordination of multimodal interactions

Content Management

Provisions for content equivalencies and dynamic filtering and translation between content formats based on environmental and personalization information and domain-specific criteria

Sensors

Sensor drivers and tools for managing sensory data

Domain-specific extensions

For each domain

- a schema developed by multi-disciplinary teams of experts in consultation with database administrators
- domain-specific views of static profile data
- tools that allow domain experts to create domain-specific workflows
- workflows will raise exceptions when required fields are not present, causing the interface configuration component to initiate user interactions that populate the missing fields
- interactions will be tailored to the end user's abilities and computing environment based on personalization information

Example – privacy with videophones

The end user's telephone is equipped with a video screen and video camera. The phone rings, and the user answers.

- The interaction configuration component consults the personalization information and security policy to determine, based on the caller ID, whether to activate the video camera or open only an audio channel

Example – Medical domain with heart sensor

The end user has sensors monitoring their heart-rate.

- The interaction configuration component manages the sensory data as it is collected.

- A domain-specific workflow is triggered by a rapid sequence of beats indicative of heart palipitations.

-The workflow, configured by the physician, takes appropriate action. This could involve

  - querying the user

  - calling the physician

  - analyzing stored sensory data for the past few minutes

  - dialing 911

Conclusion

- We are trying to create an environment that is
  - accessible (removes interaction complexity)
  - "technology pull" (configured by domain experts)
  - trusted (guarantees personal privacy)
  - adaptive
    - adapts to changes in user's needs
    - supports multimodal interactions
    - learns personalization information over time
  - peer-to-peer (ultimate flexibility for configuration)

  - human-centric