Technische Universität Darmstadt

Telecooperation

Ubiquitous & Mobile Computing

Introduction

Dr. Erwin Aitenbichler - erwin@informatik.tu-darmstadt.de
Area Head Smart Environments at Telecooperation Group
Department of Computer Science, Technische Universität Darmstadt, Germany.

Technische Universität Darmstadt

Dr. Erwin Aitenbichler
Prof. Dr. M. Mühlhäuser
Telekooperation
©

Darmstadt

Basic Information

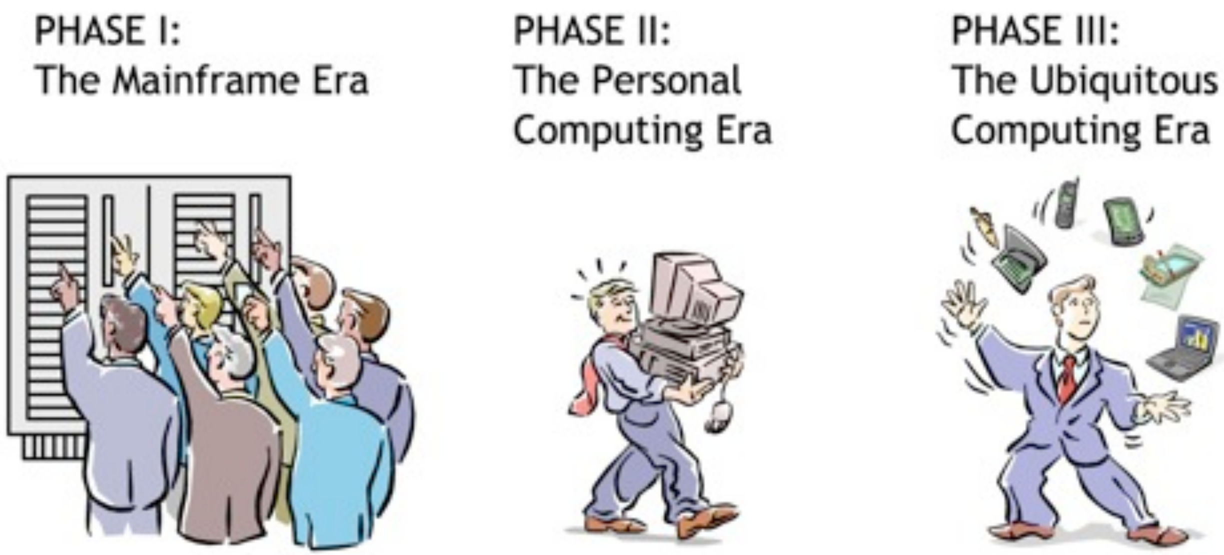
- Lectures
  - Monday afternoon (2x)
  - Tuesday morning (2x)
  - Wednesday morning (2x)
  - Thursday afternoon (2x)

- Exercises
  - Friday morning
  - Type: Introduction to Mundo tools + small Java programming exercises

Outline - What this chapter covers

- A first introduction: definition, importance
- UC challenges – a first approach: humane & integrated UC
- The S.C.A.L.E. classification of Ubiquitous Computing
- A brief history of Ubiquitous Computing (UC): Mark Weiser et al.
- From buzzword Babylon towards a taxonomy
- A Reference Architecture for Ubiquitous Computing

What is UC?



- Phase I+II: Distributed Systems
- Phase III: Ubiquitous Computing
  - Increased **heterogeneity** of devices and networks
  - Need for **integration & cooperation**
    - Trend: single user has several computing devices
    - Trend: embedded systems are no longer „closed systems“
  - **Context-awareness**

Motivation & Definition: What is UC?

- Note, for Era 3:
  - some computers are *worn* (mobile phone, PDA; body sensors, life recorder…)
  - other computers are *encountered* on-the-move, more or less consciously
    (cf. info kiosk / data projection versus badge-reader / surveillance cam)
- UC means "networked computers everywhere"
  - µC's were embedded in VCRs, washing machines etc. since *decades* already!
  - BUT: now, with UC, the embedded computers become
    - networked, i.e. part of a "web" of interacting computers surrounding us
    - "programmable": changing firmware / SW is easier, less noticeable to user
      (good: cf. bug fixes; bad: cf. undesired functionality, e.g. spying)

Definition: What is UC?

- UC connects general & special purpose computers
  - the distinction is more a spectrum than two distinct classes
  - the power / resource spectrum (CPU, memory, bandwidth, …) is *huge!*

How to 'define' Ubiquitous Computing?

A simple definition is given above: UC is the third era of computing according to the aforementioned categorization

Our more detailed and self-sustained definition:

Ubiquitous Computing is the dawning new era of computing[1], in which individuals are surrounded by many networked, spontaneously yet tightly cooperating computers,
(i) some of them worn or carried,
(ii) some of them encountered on the move,
(iii) many of them serving dedicated purposes as part of physical objects,
(iv) all of them used in an intuitive, hardly noticeable way with limited attention".

[1] following the 'mainframe' and 'personal computing' eras

Importance: Sample Nodes in UC networks

Tiny, rather general purpose computers acting, e.g., as nodes in **sensor networks** (cf. UC Berkeley's *motes*, Particle Computer GmbH's *particles*)

Wearables and mobile computing devices. cf. handhelds for ware-house picking, washable computer jackets, or companions like the "lovegetty" pictured: stores user's profile, beeps when a "compatible" person – with lovegetty – appears

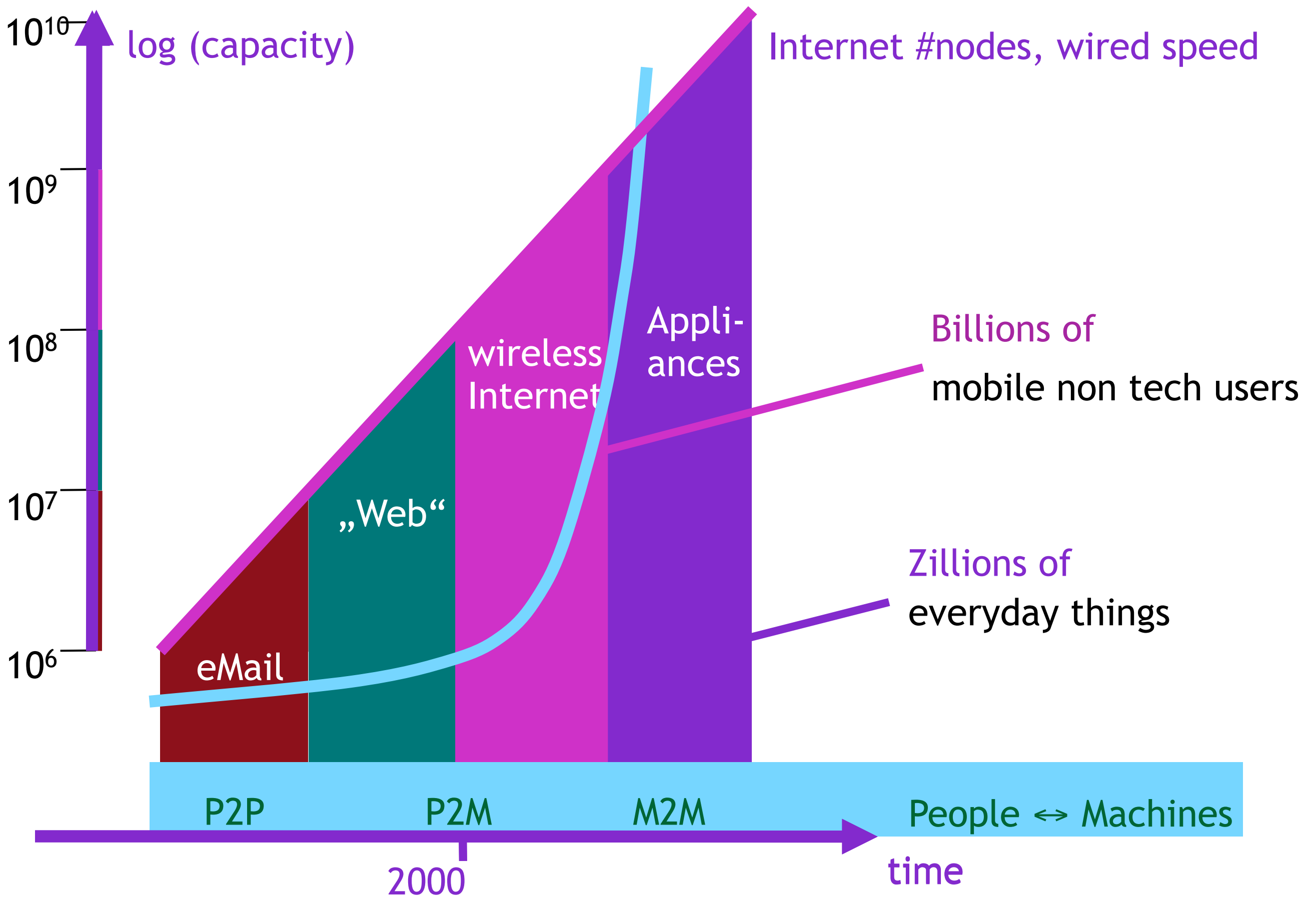
Networked appliances, e.g., "smart vending machines" that transmit fill status, errors etc. to head offices; press rumors about adaptive pricing caused protests

Smart labels such as RFIDs (above) and active badges (below) identify (option: locate, characterize)

objects, humans, animals; benefits and privacy threats widely discussed in press

Importance: Exploding # of nodes

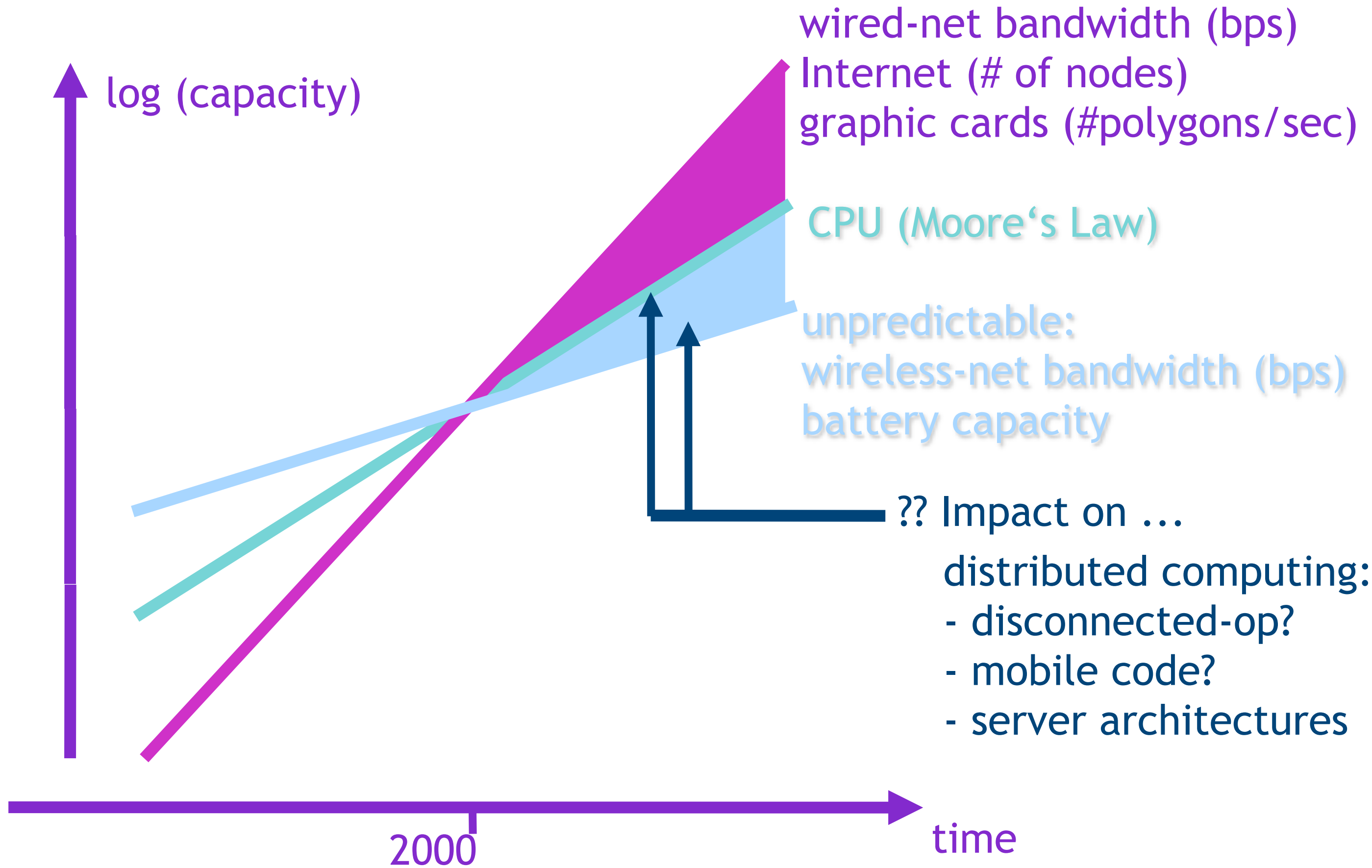
Issue #1: on the networking side, the exponential growth (# of nodes connected in the Internet) continues way beyond the world's population
consequence: scalability becomes even more crucial



Dr. Erwin Aitenbichler
Prof. Dr. M. Mühlhäuser
Telekooperation
©

Importance: exploding # of nodes, consequence

Issue #2: different doubling rates for networks, CPUs (Moore's law) etc.; impact on: system architectures, algorithms, etc. (e.g., RPC becomes cheaper, mobile computing maybe more expensive?)



Importance: Conclusion

At the 30,000 ft. level, UC is “the next key technology to change our society” for 3 reasons:

1. UC describes *the* next era of computing
   - therefore, all of computer science / IT is potentially impacted
   - … it will be difficult to be selective in designing this course/book (but we will see that a course / book on UC still makes a lot of sense)
2. UC has potential impact on every facet of our lives
   - computing is no more “what we do when we sit at the computer” + “what is encapsulated/hidden deep inside VCRs etc.”
   - see the more / less comparison on next slide
3. UC is impossible and inevitable at the same time
   - components (“gadgets”) are developed and deployed, cf. slide “UC sample nodes” above
   - more and more UC scenarios become profitable → industry will push
   - many issues of “the whole” i.e. integrated UC systems still unresolved

From Importance to Challenges

A first glance (not yet structured, non exclusive) at challenges…

Anytime/anywhere presence of networked computers means:

• more sensitivity	↔	less protection	(cf. security)
• more dependence	↔	less perfection	(cf. reliability)
• more obtrusion	↔	less attention	(cf. HCI)
• more garrulity	↔	less throughput	(cf. networking)

In other words: as computers become ubiquitous …

- there is a risk that "the whole may be way *less* than the sum of the parts" i.e. the desired integral functionality may lack way behind
- some known problems aggravate considerably – we've got to care!
- in short, UC is a problem of complex integrated systems, not of gadgets compare to: new Airbus, Space Shuttle, not better rocket-fuel, headlights

Humane & Integrated UC - Overview

features of UC „gadgets“ are not the problem ...

... but making them „humane“ and integrating them into a useful whole

Top Level Challenges:

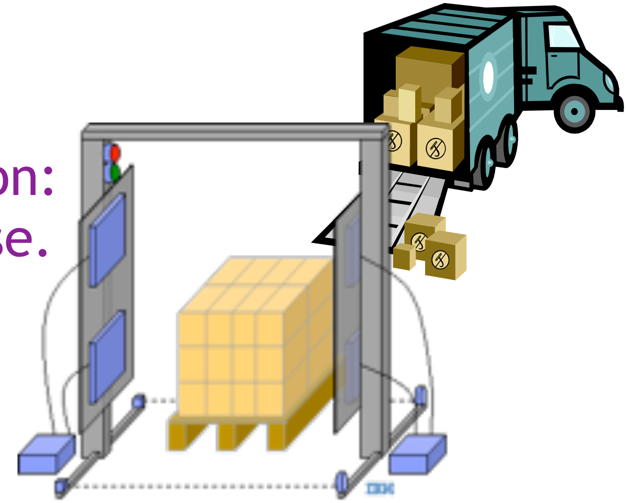
A. **Systems Integration** (Strength of Europe)

B. **Humane Computing** (Chance for Europe)

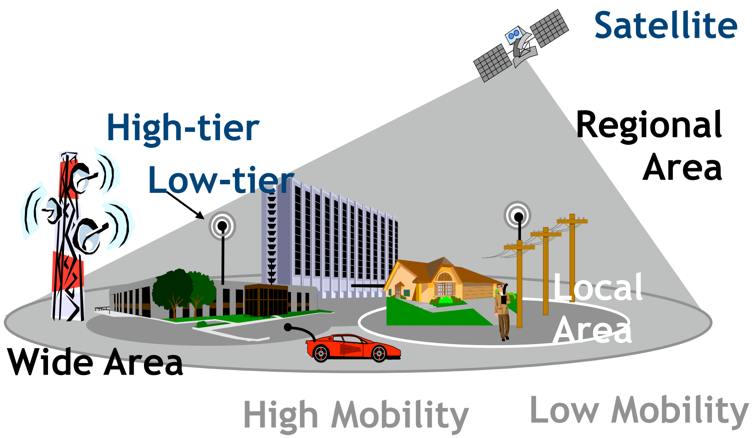
Outlook (see below):

- Scalability
- Connectivity
- Adaptability
- Liability (=Security ++)
- Ease-of-Use

Challenges: UC Integration



Pertinent Application: Real-Time Enterprise.
full coverage: production/ supply chain, retail, post-sales



Global Internet: seamless context-aware (service!) network

Challenges: Humane UC

Mainstream UC Research emphasizes two issues:

- Context-Aware Computing

  ill defined; basically, the quest for adaptive software that …

  - understands the „situation in which the user currently works“
  - adapts to that situation by
    - *reducing* complexity (e.g., adjusting UI to current needs & options)
    - *automating* input (e.g., retrieving data from „environment“, not user)
    - *optimizing* functionality

- Multiple Modalities

  again, ill defined; modality is a „way of doing I/O“, often an input and/or output „channel“ (what is that?) like graphics, voice, keyboard/mouse, gesture (note: examples from different categories!)

  - UC→ applications on appliances (cell phone, PDA… today, more tomorrow)
  - → Multiple modalities to be supported
  - truly multimodal UI: flexible combination of „channels“, user‘s choice

S.C.A.L.E. Overview: Breadth vs. Depth

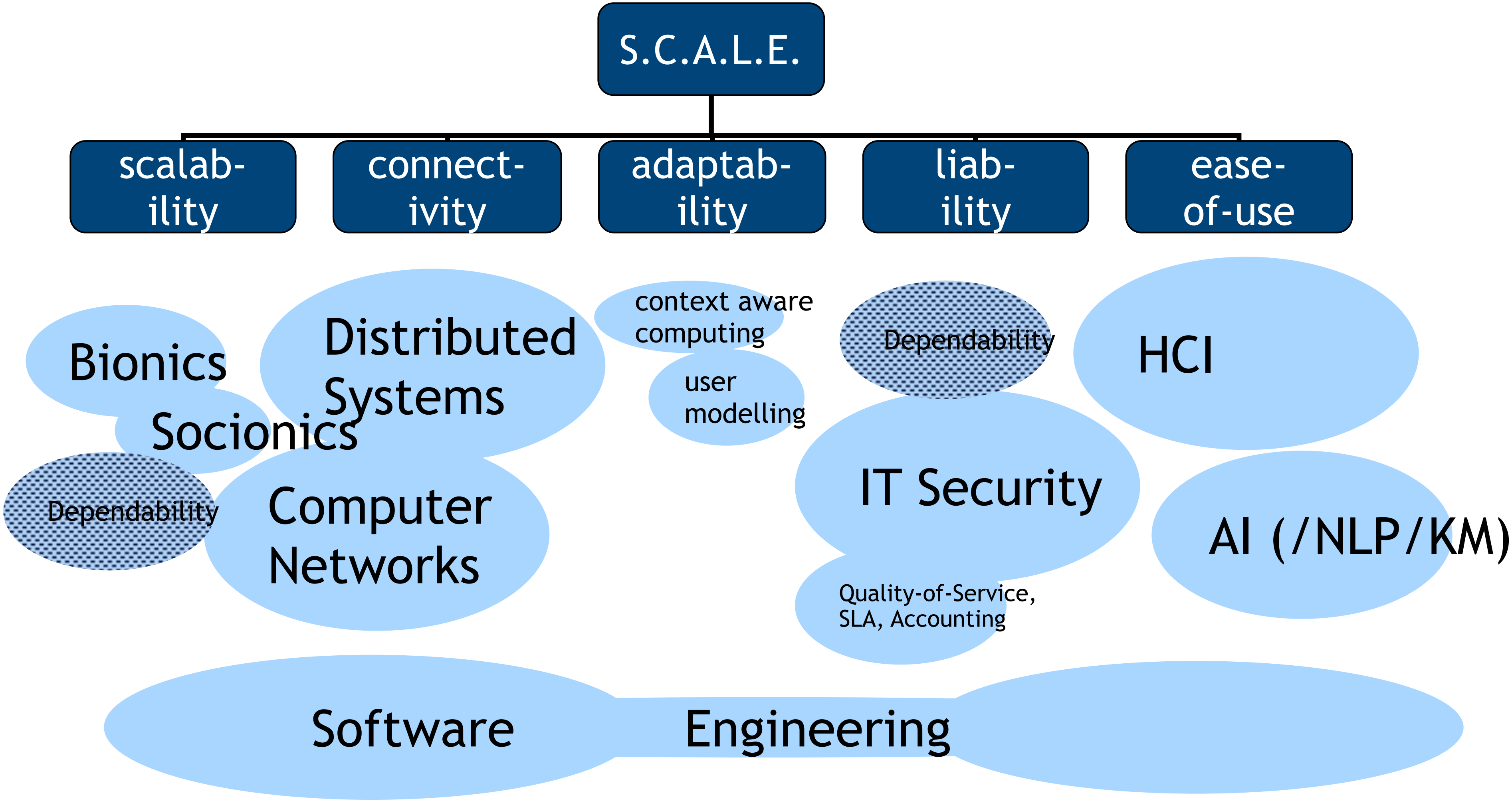
- Problem: UC touches almost every area of Computer Science, and more!
  - readers need background for understanding why what will be emphasized later
  - a categorization of issues (taxonomy) is needed for organizing the topics and for getting oriented → cf. the S.C.A.L.E. taxonomy
- So, is it a research/teaching subject in its own right?
  - yes, compare to Distributed Systems:
    - there is distributed simulation, programming, algorithms, databases, AI, …
    - yet, there are courses on Distributed Systems
  - but: how to master breadth vs. depth?
    - be broad first, organize "problem space" → S.C.A.L.E., see below
    - then emphasize pertinent problems (new in UbiComp or much more relevant)

S.C.A.L.E. Scheme

Major UC challenges, on a very high level:

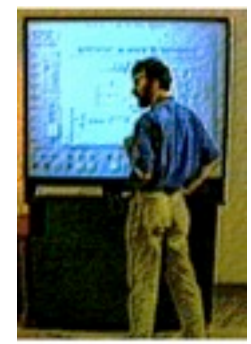
- S – SCALABILITY
  - how to support cooperation of "zillions" of components?
  - how to support nomadic users around the globe?
- C – CONNECTIVITY
  - how to "easily" connect these zillions?
  - wireless networks – a blessing and a curse (unreliable!); important but...
  - most issues *above* wired/wireless net (note: overlaps scalability): how to find/understand peers, enable zero configuration, design huge networks w/o server/bottleneck
- A – ADAPTABILITY
  - usage during daily work, surrounded by 100s of components: need *minimal* interaction
  - major approach: context-aware computing – use it to automate tasks & reduce options
  - adapting-to-user (user *modeling)* must be focused beyond context-awareness
- L- LIABILITY
  - term indicates: we must go beyond today's IT security solutions (not goals)
  - today's solutions do not scale (root PKI?), are not "humane"
  - & don't flexibly consider conflicting (privacy, traceability) / related goals (dependability...)
- E – EASE-OF-USE
  - adaptability permits "minimal" ..., ease-of-use means "optimal" interaction (related!)
  - issue: optimal use & combination of modalities, advancement of specific modalities
  - issue: "understanding" natural input: a) xxx-to-text; b) text-to-meaning; "intelligence"?

Relation to Computer Science Research



History: Mark Weiser

- *1952, †1999 (died before UC really took off)
- Coined the term, spread the vision of UC
- Most famous: article in *Scientific American:*
  The Computer of the 21st Century (1991)
  Web pre-version cf. http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html
- Worked at Xerox PARC (now: PARC), *the* (?)
  world leading center of research combining
  computing with humanities (≈ “birthplace” of
  mouse, windows-based UIs, desktop metaphor,
  laser printer, many CSCW contributions and much more)
- Co-Developed prototype UC devices, in particular 3:
  - Pad, a prototype PDA
  - Tab, a prototype TabletPC
  - Liveboard, cf. Smartboards™ --- all in the late 80es!!



Xerox Liveboard

he investigated their *integration* in group work (CSCW) scenarios, imagined the ubiquitous availability of Tabs (laying around in meeting rooms, personalized as users grab them), …

History: Mark Weiser

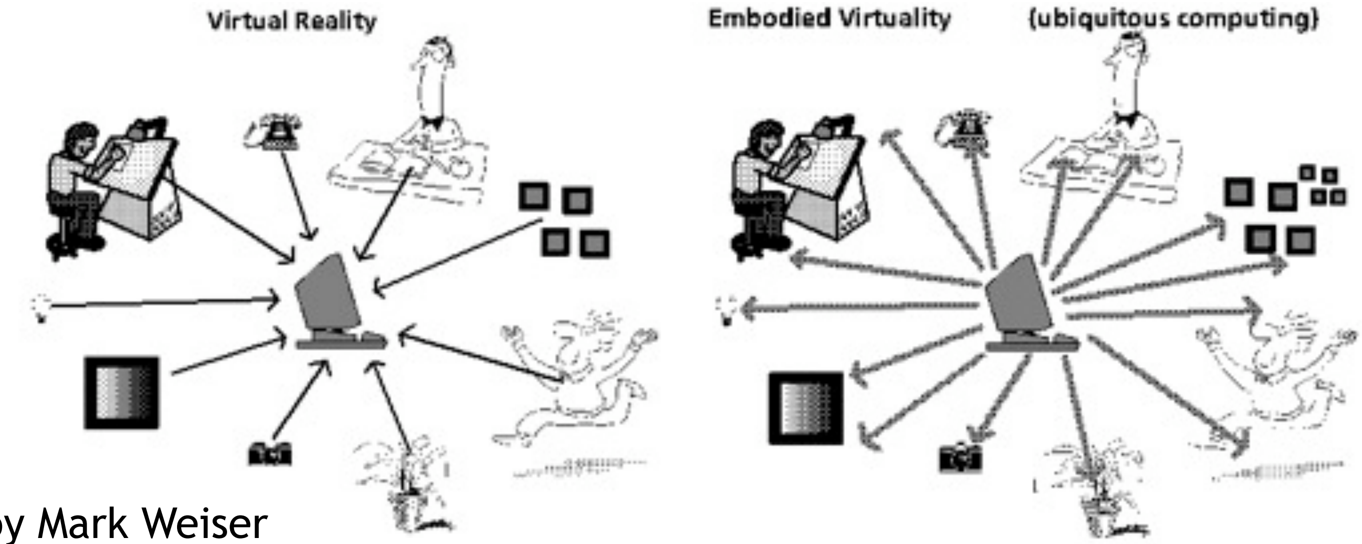
- MW's most frequently cited statement: *"The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it."*
- But MW's vision comprised more, see below
- We will try to recall Mark's view on three conflicting issues:
  - UC vs. VR (virtual reality)
  - UC vs. AI (artificial intelligence)
  - UC vs. UA (user agents)
- Thereby, we imagine Mark at panel discussions, as a UC advocat trying to argue why VR, AI, and UA are "dead end" research roads while UC is the open road
- Later, we will take a less dramatic standpoint

History: Mark Weiser – UC vs. VR

VR is based on (3D *and* semantic) models of the real world

- (ever larger, ever more detailed) cut-out of the world is modeled in the computer
- put to an extreme: the world is moved into the computer
- … and even the user becomes a computer peripheral (hmd, data glove)
- with UC, in contrast, the computer is moved into the world!!

→ not one "big boss" computer, but many small ones with dedicated task & responsibility
→ networking for making sense of the small parts
→ *not the computer is in the center, but the human!*



cartoons by Mark Weiser

History: Mark Weiser – UC vs. AI

- around 1980, AI had been over-hyped
- around 1990, frustration reigned: AI had not lived up to its promise
- Mark's argument: analogy computer = brain is exaggerated
  - terms "intelligent" and "knowledge" raise too many expectations
  - the AI vision of intelligence concentrated in a computer is wrong
- With UC, in contrast, we aim at *smart* components
  - they figure out a tiny cut-out of the world only (just temperature or just presence of object, at just a small location, …)
  - smart computers compare to intelligent computers like neurons to brains
  - higher-level "sense" comes from networking smart components

History: Mark Weiser – UC vs. UA

- in contrast to VR and AI, user agents are not very prominent any more; some basic concepts have remained though
- UAs were thought as intelligent intermediaries between the user and the computer world, thus an approach towards ease-of-use / HCI
- Mark challenged 5 requirements for UA as “dead end roads”:
  1. UAs should give advise --- why don’t they do the job themselves?
  2. UAs should obey (like a butler) --- why aren’t they more proactive?
  3. UAs should work at the interface --- why interact and not *do* things
  4. UAs should listen to the user --- with immature natural language processing technology, speech recognition etc., how should they understand?
  5. UAs should learn the user’s preferences, wishes etc. by observing --- with immature machine learning technology, how should they do the right thing
- UC, in contrast, according to Mark, should aim at “agents” which
  - carry out actions and not just mediate
  - do that largely autonomously such as not to bother the user
  - … and therefore have not much of an interface at all

History: Two more visionaries

Don Norman's book „The Invisible Computer“ (1999)

- critique: PCs are complex, try to be all-purpose/all-user
- critique: PCs are isolated from daily work & live

therefore, Don is the first one to clearly demand:

- Information appliances: dedicated for specific task/problem → way simpler and more optimized
- Human-Centered Development: design the appliance such as to optimally support its user
- design axioms simplicity, versatility, & pleasurability
- „systems“ to be flexibly composed families of appliances

History: Two more visionaries

Kevin Kelly's book "Out of Control" (1994):

- the complexity of the made (engineering) reaches the complexity of the born (cf. biology/nature, social organisms)
- we should "learn the principles" of "the born" and adopt
- looks at bee hives, evolution, and many more: how do they cope with errors? with change? with control? …
- looks at industrial evolution, the quest for perfection
- proposes things like:
  - give away control (autonomous responsible behavior of part in whole)
  - accept errors for selection, adaption, constant optimization
  - truly distribute control (no central instance)
  - promote chunks (hierarchies …) for taming complexity
  - accept heterogeneity

History vs. Present

Mark Weiser was not perfect (like any visionary). Recall his 3 points:

- VR-vs.-UC (embodied virtuality) dispute – today, we would say:
  - we need „the computer in the world“ …
  - but also „the world (model, distributed) in the computers“ if the cooperating whole shall make sense of the „smart“ parts
  - (VR approaches this reconciled view with augmented reality (AR) concepts)
- AI-vs.-UC dispute:
  - AI was indeed overhyped …
  - but we need to find out how to combine „smart computers“ like neurons into a „brain“ that makes sense
  - note: similar argument as above; conclusion: *integration is key*
- UA-vs.-UC dispute:
  - „listen to user and learn“ was (over?) ambitious …
  - but (1) autonomous-actions instead of „obey/advice“: even more ambitious
  - but (2) machine learning made progress → „learn->advice“ becomes feasible

History vs. Present

Kevin Kelly

- autonomic computing and many „soft computing“ disciplines (neural networks, ant colonies, evolutionary algorithms) follow his principles
- … but scalability *plus* reliability remain a huge challenge
- again: *integration is key*

Don Norman & Mark Weiser

- user-centered design, context-aware computing, multimodal UIs etc. follow their quest for „humans in the center“
- … but „humane“ computers comprise further issues, in particular UC-ready IT security (cf. F. Stoyano‘s book on „Ubiquitous Security“)
- in summary: *humane computing is key*

Name of Discipline

UC has many names (sigh! hinders establishment as wide-spread discipline!)

- **Ubiquitous Computing:** more common in academia
  emphasizes the „final“ state of penetration: computers everywhere
- **Pervasive Computing:** more common in industry
  (coined by IBM?); emphasizes computers „penetrating“ the world
- **Ambient Intelligence:** „invented“ for EU research framework programs (5, 6, 7);
  „ambient“ refers to Weiser‘s quote, „disappearing in the environment“;
  „intelligent“ is revival of over-hyped term → Amb.I. is *only* common in Europe
- „Hypothesis“:

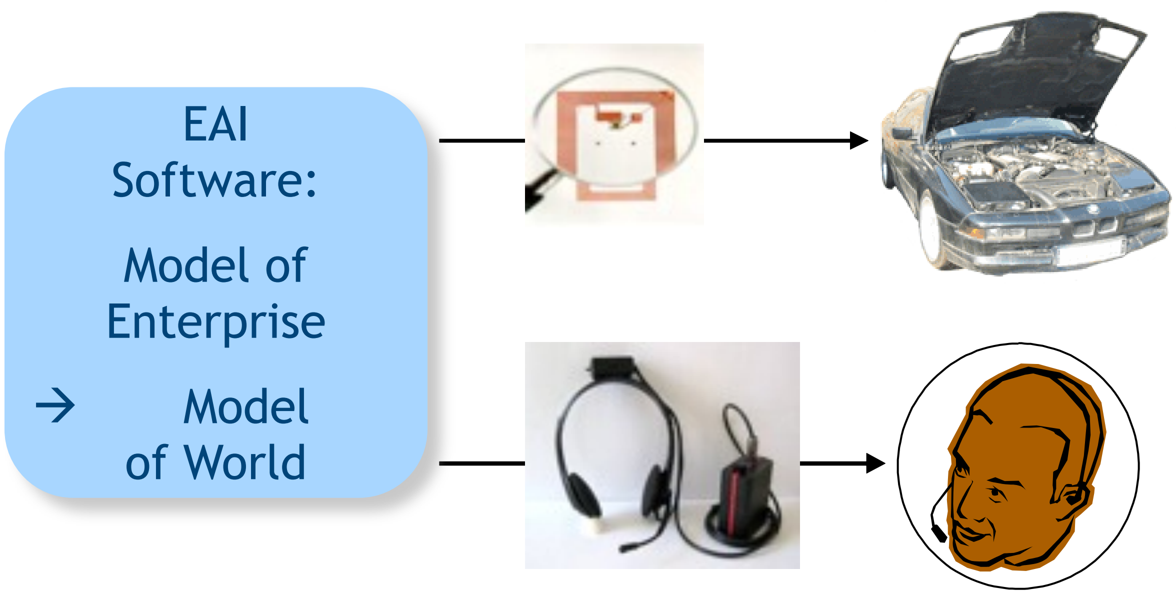
  Ubiquitous Computing = Pervasive Computing = Ambient Intelligence
  Google-Hits: 870k 670k 410k (circa)
- more, less commonly used terms: post-PC era, disappearing computer, calm computing, sentient computing, mixed-mode systems, tangible bits, real-time enterprises

Need-to-Know: Real-time Enterprises

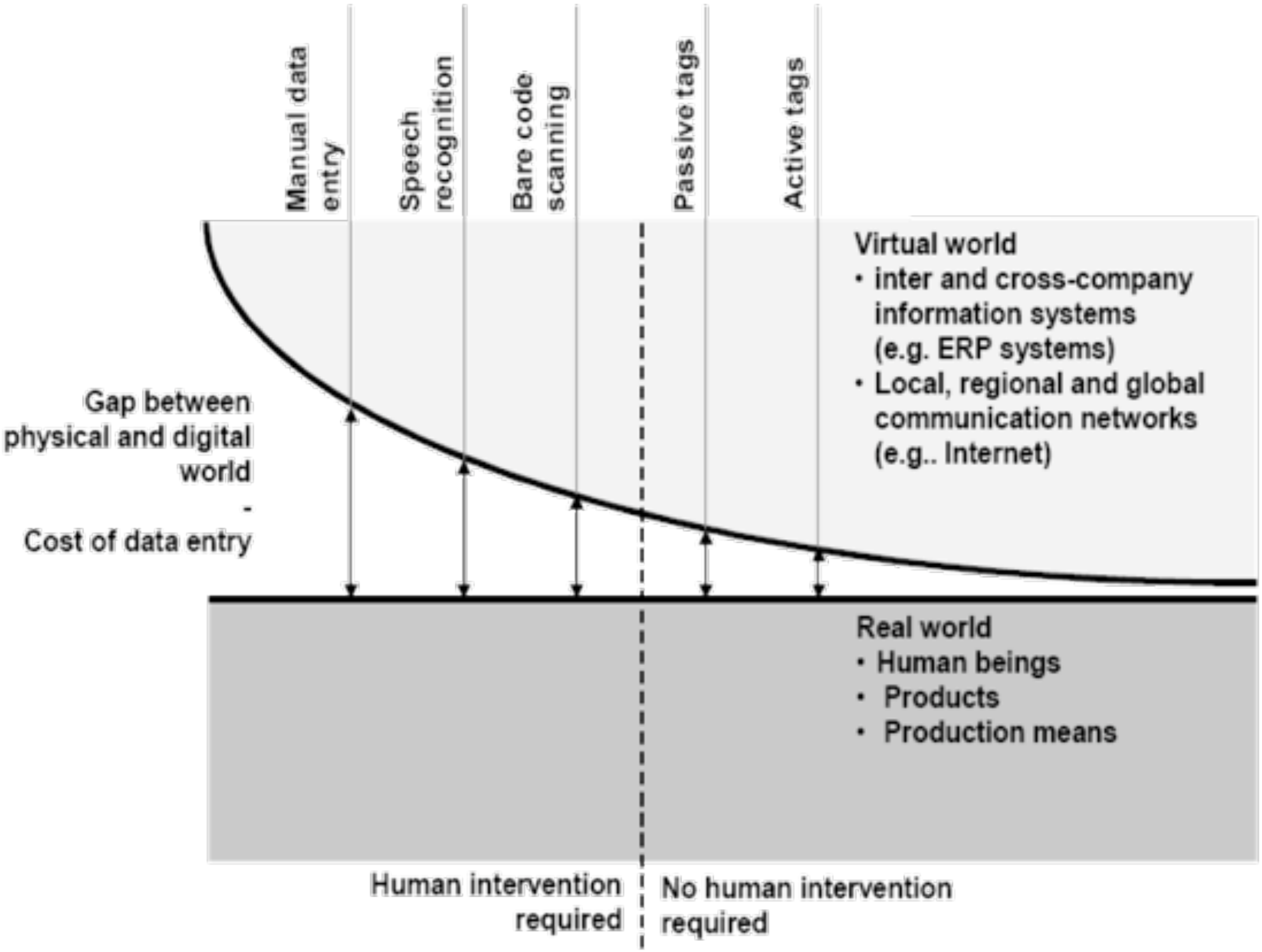
Real time enterprise (RTE): used in enterprise software context

RTE reduces the gap (see next slide) between

- model of the world in the computer (note difference to MarkWeiser-View!!) and
- real world itself, by bringing computer to the world (today mainly: via RFID tags)
- RTE has on line:
  - ✓ spaces (environment): via sensors, actuators
    & items: via tags (cf. RFID) → embedded (Internet) appliances
  - humans: mostly hands-free / eyes-free

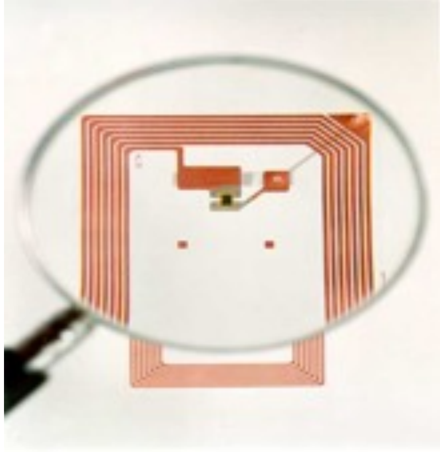


Need-to-Know: Real-time Enterprises



Need-to-Know: Internet of Things

- The Internet of Things: term favoured by press
  - emphasizes 50% of ubiquitous computing (cf. taxonomy): “encountered”
  - discussion dominated by emerging smart tags standards, in particular AutoID
  - … and by Internet appliance standards, in particular OSGi
- AutoID: center at M.I.T. standardizing RFID-based successor of “Barcode”
  - first, a look at RFID chips:
    - actual chip may be only 4 mm^2
    - „giant“ antenna
    - on chip: ID-no. „burnt into ROM“
    - may contain RAM
    - … and active communication (, even CPU)
    - for AUTO-ID: ID is 96-Bit electronic product code EPC
    - EPC succeeds barcode, has serial-no!
    - compare to “class ID” vs. “object ID”
    - for mass markets (Walmart etc.), currently used mainly for “palette & case level tagging”, not for “item level tagging” yet
  - Generation 2 (2006): improvements
    - smaller, cheaper
    - reader reads hundreds of tags “simultaneously”?
    - “printable”: paper lables with embedded RFID?



Need-to-Know: Smart Items

- Introduced by UCB, UCLA; now in Europe (SmartIt’s etc.), …
- Small, resource limited devices
  - CPU, disk, power, bandwidth, etc.
  - witt simple scalar sensors – temperature, motion
- some: customized to single domain /task (ecology … health … military)
- ad-hoc wireless network (ZigBee or private, wLAN? power hungry!)
- ingredients, e.g., Atmel CPU, TinyOS, TinyDB, power conservation …
- e.g., Berkeley Motes




Need-to-Know: Smart Homes

Smart Home := Smart Environment Category: Your Home

- many projects worldwide
- some prestigeous projects
  - industry: Microsoft eHome, Philips AmbientIntelligence, …
  - academia: GeorgiaTech AwareHome, MIT House_n, …
  - platforms: HP Cooltown
- but many projects “terminated”: business case? user acceptance?
  - current hope: business case *assisted living (cf. AgingSocieties like Europe9*
    - e.g., keeping elderly selfsustained for 1 more year in life: saves ~.5 bn € in Germany alone?
    - in this area, projects with large user studies: Zwijndrecht (B), Tønsberg (N), …
  - other hopes: home security, energy conservation, home entertainment
- BUT – Bottomline:
  - high likelyhood that large scale deployment is in business, not home!
    - e.g., logistics, inspection, manufacturing, services …
    - why? business cases more obvious: companies invest! (heavily)

Need-to-Know: Further Buzzwords

Above taxonomy is helpful, will be used in the remainder …

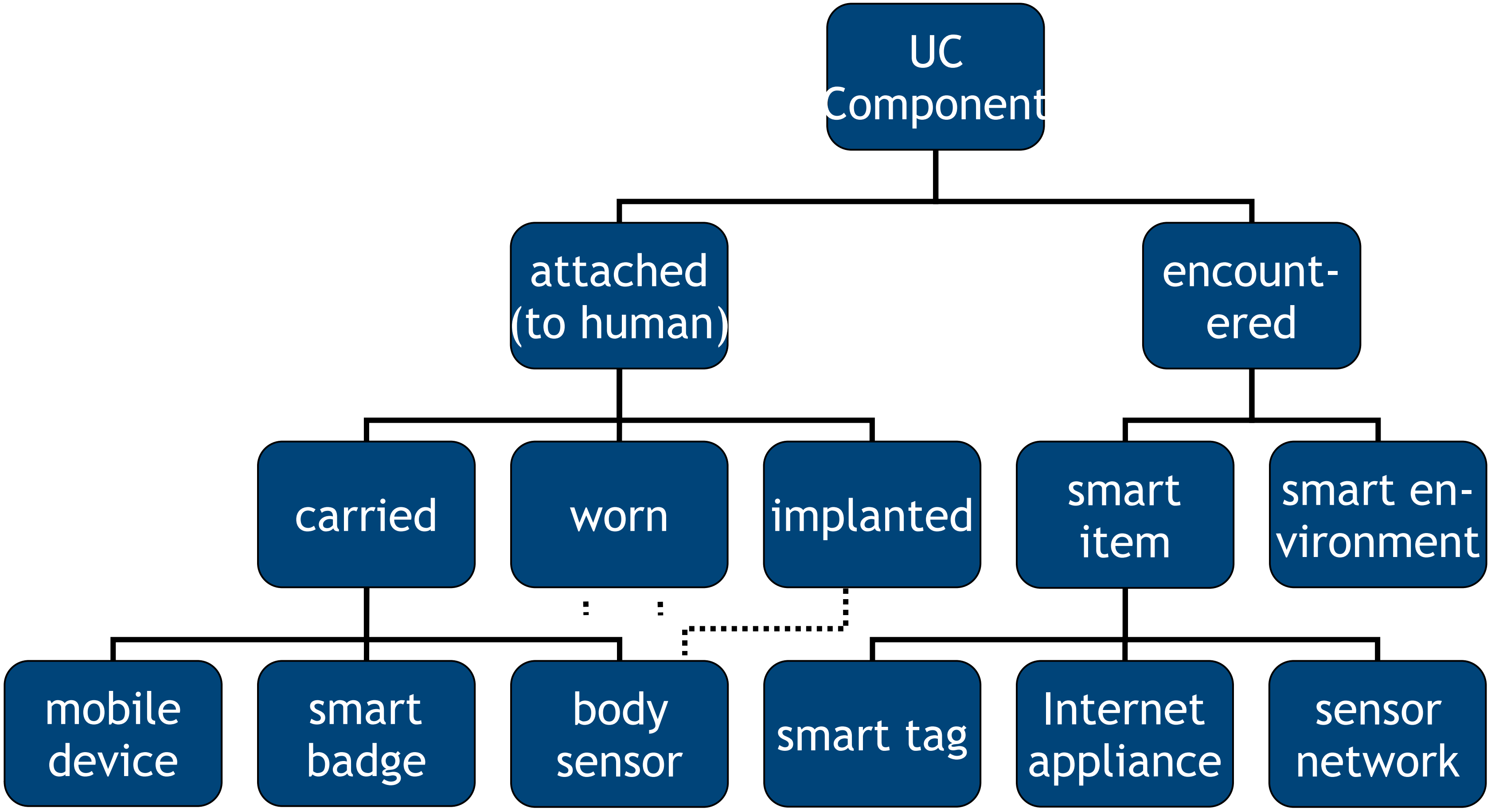
- … but: not settled → other buzzwords cannot be fully classified yet
- like, above: federated …, smart objects/products/spaces (also: see our *own* definition of smart products, later)

Further UC components to be mentioned, but not fully classified yet:

- Smart Dust: alludes to zillions of *very small* sensors
  - vision 1: auto-decay (organic? → compost ☺)
  - vision 2: edible (health examination etc.) → inhalable?
    - variant: picked up "afterwards", data read out → not truly a network
  - feasible today (example): aircraft sheds sensors over contaminated area, sensors cooperate → deliver environment data
- Things-that-think (project, N. Negroponte, MIT media lab)
  slogan: "*in the past, shoes could **stink**. In the present, shoes can **blink**. In the future, shoes will **think***"
- Smart Paper (new category: smart materials?)
  - originally: re-usable carrier for "daily news" etc.
  - today, term sometimes misused by press/marketing

Taxonomy: Smart Items

We may start to organize UC components in a real taxonomy:

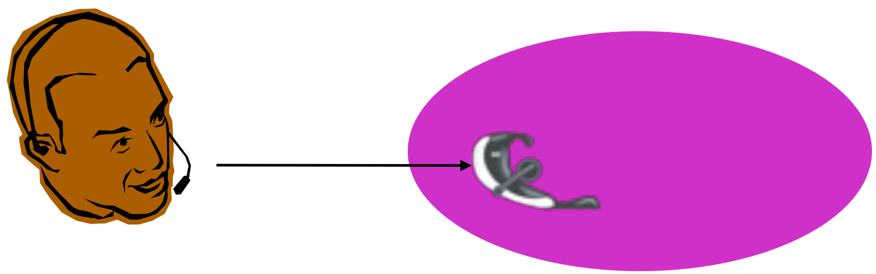


Reference Architecture: Motivation

Why a reference architecture?

- "Old definition" of Distributed Systems not appropriate
  - OLD: "a distributed system is a collection of autonomous systems AS, interconnected through a communication subsystem CSS": DS :== {AS} ∪ CSS …
  - … where communication happens via message exchange only
    - note 1: an AS maybe multi processor with shared memory
    - note 2: an AS was defined as possessing:
      1. processor, 2. memory, 3. communication, 4. identity (e.g., IP address)
- Futher "old" views:
  - client-server world: two kinds of nodes exist (clients, servers), maybe blurred
  - peer2peer world: deliberately no distinction between nodes
- However, in UC world, there are 2 main reasons to introduce distinctions:
  - resource heterogeneity: special-purpose inappropriate for general-purpose tasks
  - role heterogeneity: very personal nodes should not be treated like very public ones?
- Component Architecture: describes interacting components
  - Mundo Reference Architecture by TUD

Mundo Reference Architecture



Mundo departs from the following assumption:

- UC makes everyday life “computer assisted”
- → many actions will be machine2machine, many of them representing user’s will / preference / general-permission
  AND: these actions will have legal impacts (cost, liability, …)
- → acceptance is only feasible with a “digital persona” in which the user has full trust
- digital persona: a digital representative of the users, acting on her behalf
  - authentication: User ⇨ ME (once), ME ⇨ Environment (many times)
- we call this:

ME = **M**inimal **E**ntity

Mundo Reference Architecture

Requirements for a ME:

- full trust by user: device must be understandable, “under control”
- in order not to rely on other devices (such as a USB stick relying on a Laptop for PIN entry!), the ME must possess:

  1. CPU, 2. Memory, 3. User I/O, 4. (Wireless) Network I/O

- for reasonable minimal functionality, it is useful to add

  5. context-sensitivity (location, head orientation found useful!)

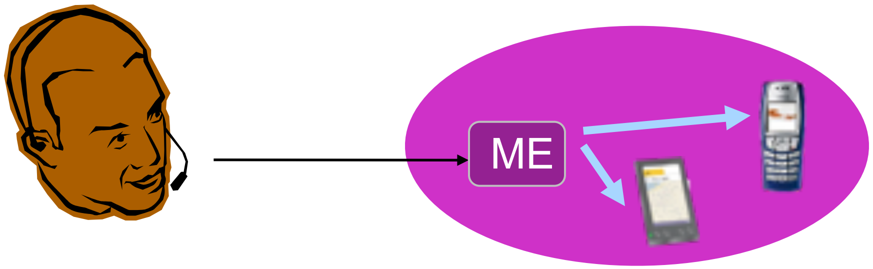
- as a fully trusted party, it must be able to act as a

  6. Persona Support capable of providing Ubiquitous Security; this functionality should be bootstrapped via highly secure credentials (e.g., fingerprint)

User I/O:

- the device is to be carried at all times (since it accompanies everyday actions)
- interaction will often happen in limited-attention, hands-/eyes-busy mode
- therefore, voice interaction appears to be a natural choice
  - complemented by touch/gesture in silent & very noisy environment
  - complemented by the full range of I/O via device association, see below

Mundo Reference Architecture



Other devices to be operated on the user's behalf are granted functionality & controlled *through* the ME: they are *associated* with the user

This leads to a second class of components:

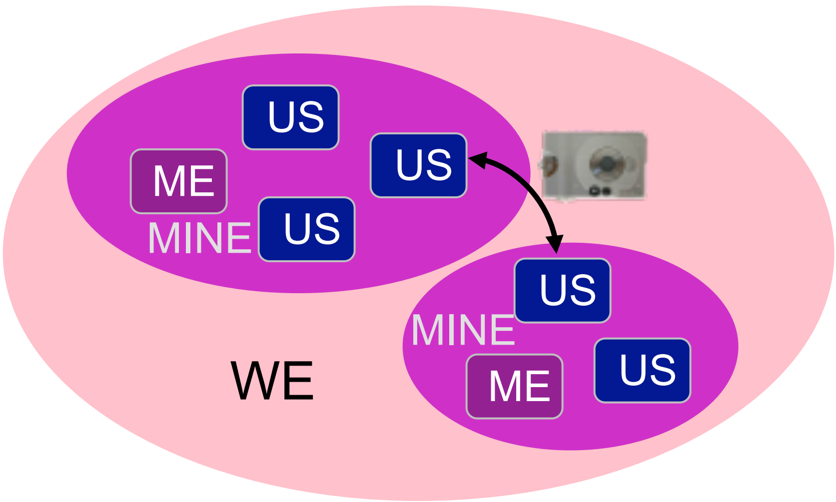
US = User aSsociable

Secure protocols for temporarily associating particular devices have been developed;

ME and US form a trusted PicoNet named:

MINE = Mundo Integrated Network Environment

Mundo Reference Architecture



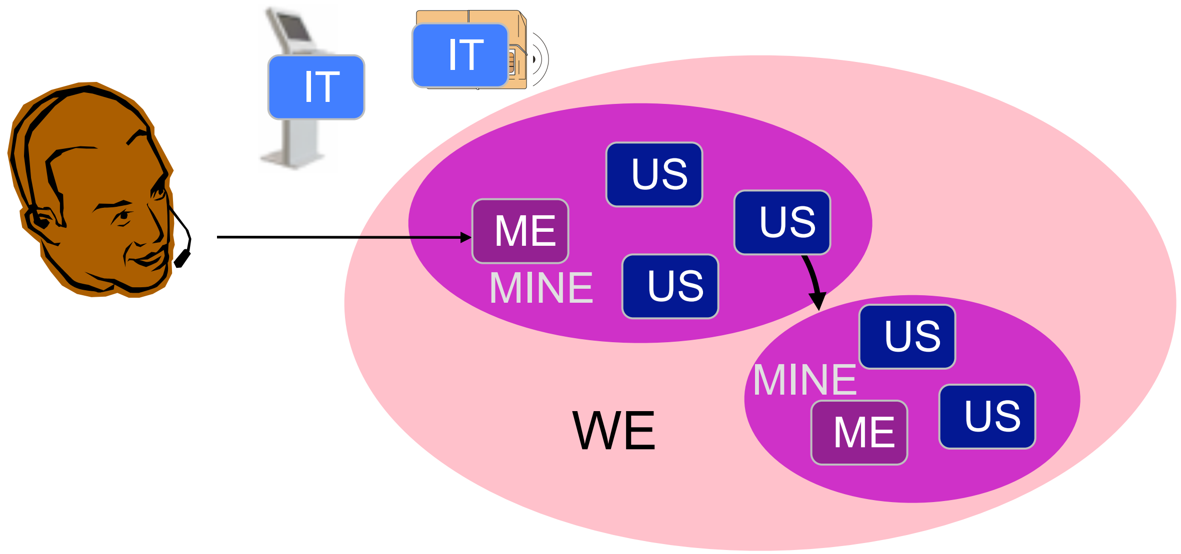
As several users with MEs and USes meet, they may be supported in forming a

WE = **W**ireless group **E**nvironment

Association of devices may change (e.g., think of one user lending her DigiCam to another user: during re-association, the old user's photos may be hidden, the temporary user's photos may be send to the appropriate storage, …)

More on “WE” type configurations: see chapter on Opportunistic Networking

Mundo Reference Architecture



Devices encountered on the move *may* be associable (→ turn into USes),

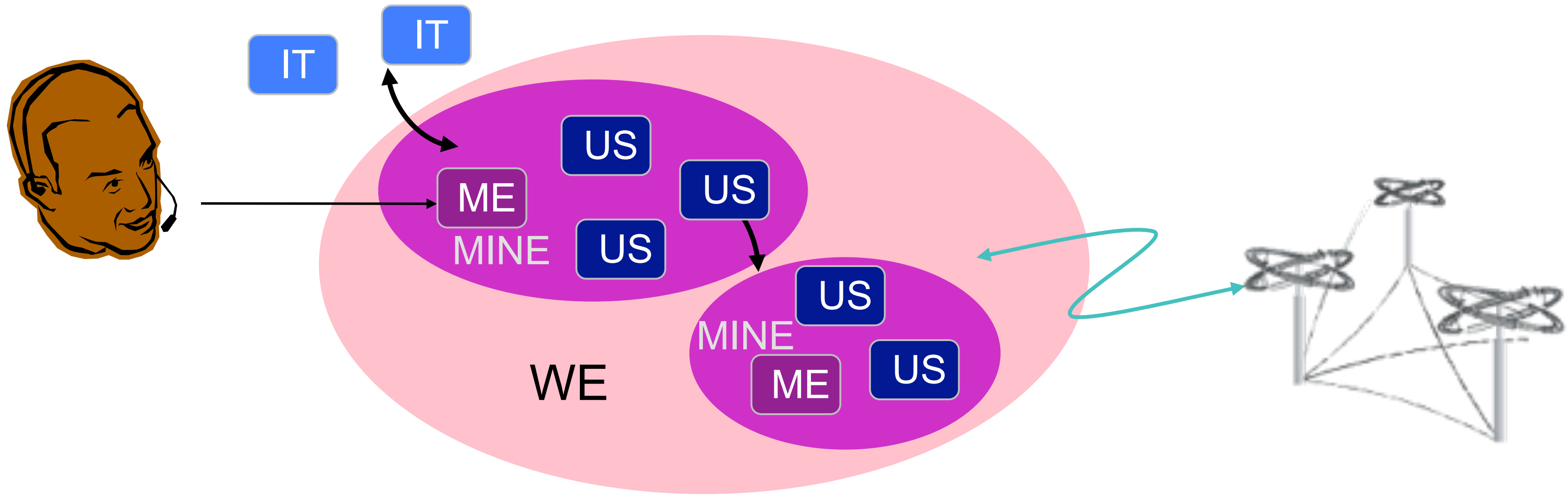
but they may also lack full association capabilities.

Rather, they may deliberately by designed for cooperative or shared use Such components are called

IT = smart Item

For the user, the US – IT distinction is crucial in terms of liability!

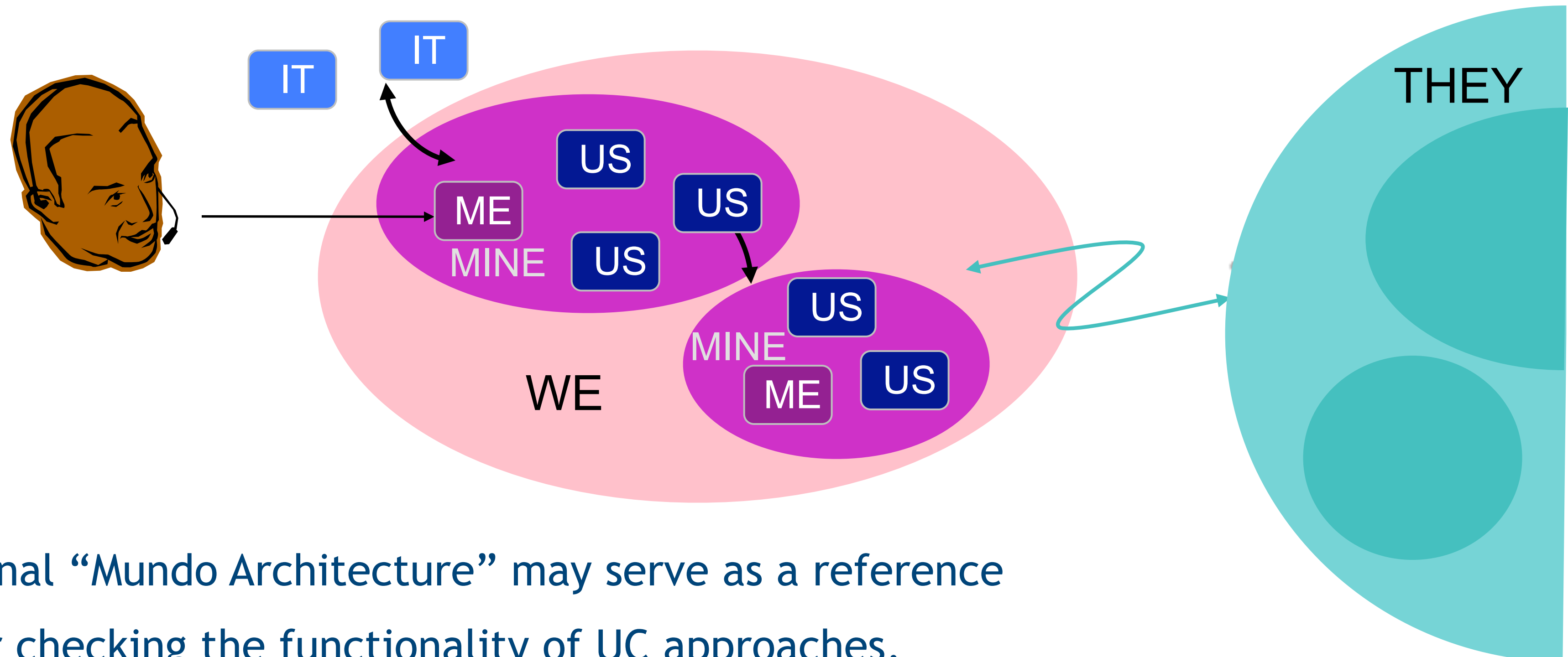
Mundo Reference Architecture



Finally (and maybe most important), functionality will be provided through "intangible" services in the network; since such services will be provided in overlay networks of all kinds (see corresponding chapters), the encompassing set of such services is denoted as

THEY = Telecooperative Hierarchical ovErlaY networks

Mundo Reference Architecture



The final “Mundo Architecture” may serve as a reference

- for checking the functionality of UC approaches, Smart Environment platforms, etc., for necessities & completeness
- for discussing required functionality of its elements (cf. the discussion about “ME” functionality above)
- for furthering the architecture as experience with UC systems grows
- for anchoring the UC issues and topics, e.g., as covered in the following

Other Reference Architectures

3 architectures (out of many) worth mentioning:

- OSI ODP (Open Distributed Processing): mix of layered/component
  - rather old, little success
  - tried to tame the complexity of large distributed SW systems
  - introduced views for different stakeholders (developer, user, …)
  - Follow-on: OSDI (open distr. services infrastr., rather component than layered), influenced some of the UbiComp-Archiectures built recently
- “I-Centric Services” by Fraunhofer Fokus Berlin (→ OMG, …)
  - shares with Mundo: concept of user-centric (“ME”) approach
  - yet, based on concept of “the” universal component interface: called Super Distributed Object (SDO): has standard interfaces for:
    - discovery // maintainance // reservation // configuration

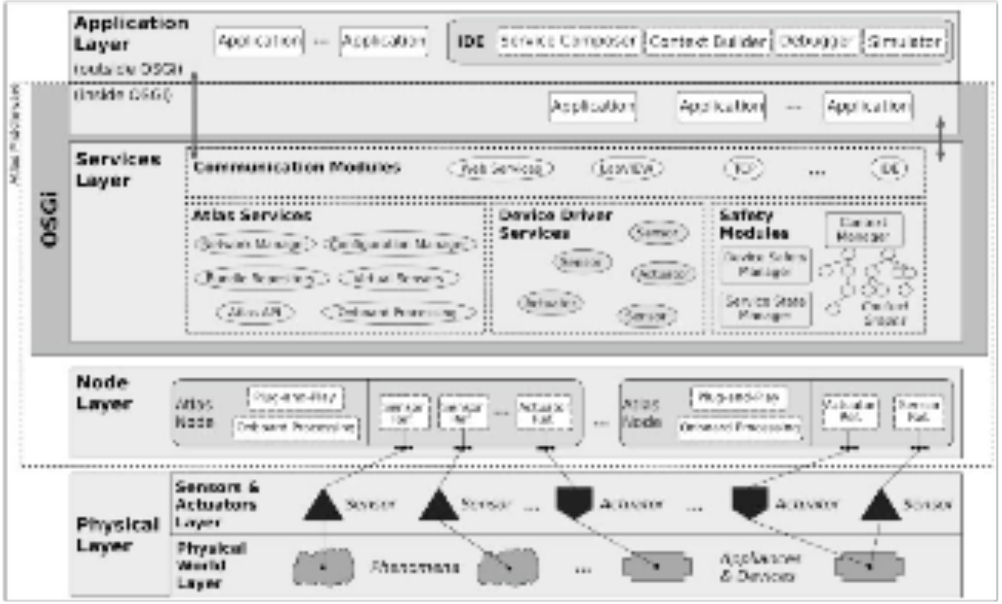
    plus further custom interfaces

Gator Tech / Atlas Reference Architecture

Gator Tech: Prof. Helal & Groudn: www.icta.ufl.edu; Atlas company: www.pervasa.com

- Emphasis: sensors (+actuators) & networked embedded devices - layer 1
- OSGI exploited for customizing and maintaining L.1 Hardware in dedicated L.2
- L.3 contains 3 large parts (cf. UC 'world'):
  - *context management layer*
  - *service layer:* SOA dominating paradigm for autonomous software components in UC
  - *knowledge layer:* reflects problems of large-scale service composition: service discovery & interaction via machine readable descriptions of service semantics @ runtime;
- strict SoA concept → appl. development = service composition → *top layer:* composition tools

Atlas architecture =
"modified" GatorTech:



Introduction: For Your Long Term Memory

Ubiquitous Computing:

- smart devices, roaming users, zillions of nodes
- inevitable, even reality today…
- … but integration remains a grand challenge
- Challenge as five large research issues: S.C.A.L.E.
  - scalability
  - connectivity
  - adaptability
  - liability
  - ease-of-use
- what else to remember:
  - the many termy & buzzwords ☹
  - reference architecture and “ideas behind” visions, organizations

Scalability

- Term became common in Computer Networks / Telecomm., Distributed Systems, and Software Engineering
- Formal definition? Only possible for narrow purpose → remain informal
- First approach: scalability is the ability to grow
  - ability of what? of a "*solution* to (a set of, certain) *problems*": algorithm, method, system (software system, platform, …), network, …
  - ability implies: "… grow and remain *useful*" … in the performance sense
    - as 'problem' grows, resources required for 'solution' should not grow faster
    - we call this: *mathematical scalability*
  - ability also implies: "… grow and remain *usable*" in terms of "cost", acceptance
    - acceptable effort to add / maintain / remove components
    - acceptable effort to migrate / re-use components & systems
    - acceptable usability (for humans, organization, …) as 'problem' grows?
    - we call this: *technical scalability*
  - term scalability is also used in economic / commercial world
    - cf. http://en.wikipedia.org/wiki/Scalability
    - means: a business is ready for growth (geographically, wrt. turnover …)
    - we call this: *economical scalability*
    - *relevant below: IT related i.e. software related issues of economical scalability*

Mathematical Scalability

More on mathematical scalability:

- for sequential programs/algorithms: *(1) computational complexity theory* (deals with relative computational difficulty of computable functions; terms like: Big-O notation, e.g., “O(N log N)”, NP-complete, etc.)
- for issues of Computer Networks / Distributed Systems:
  - *(2) Speedup* treats possible parallelization on multiple processors / nodes
    - Amdahl’s Law frustrating: divides program into parallelizable portion $p \in (0,1)$ and portion s that requires sequential computation, $s + p = 1$ (i.e.: $p = 1-s$)
      - on N processors, p can be computed in time p/N; overall time: s + p/N
      - since s+p=1, yields speedup (1 vs. N processors):
      - basically bound by 1/s
      - e.g., s=4% → speedup bound to 25fold (only!!)

$$\rightarrow \frac{s+p}{s+\frac{p}{N}} = \frac{1}{s+\frac{1-s}{N}}$$

    - Gustafson’s law: “counter attack”, much more promising results, cf. http://www.scl.ameslab.gov/Publications/Gus/AmdahlsLaw/Amdahls.html
  - (3) **CCCDS:** computational & communication complexity in distributed systems (no common term!) is concerned with growing “populations” in network
    - assume growth of problem size (e.g., no. of users or processes joining network fabric) comes with growth of resources (e.g., users join network with “their” PC & cable)
    - if every node has to keep track of every other node: won’t scale!
    - often: tradeoff computation - #of messages – size of messages

Mathematical Scalability

More on mathematical scalability, contd.:

CCCDS, contd.:

- problem 1 - meshing: if 'added user' means 'added node' (PC), then
  - resources of type "CPU, memory" grow at same scale as 'problem' (users), ***but***
  - connectivity i.e. no. of average adjacent nodes does *not* keep pace: new PC adds 1 link → only direct neighbor has 1 add'l connection
  - this issue was coined as *Flatland* (referring to a 19th century novella by E.A. Abbott)
  - a quest was made to 'escape from Flatland'
  - this lead to the *Hypercube*, a hierarchical multiprocessor architecture where max. distance (no. of hops) between N nodes is O(log N) – at the price of O(log N) links added for each additional node
    - in the Internet, hierarchies of (sub)networks and routers play a similar role
- problem 2 – explosion: if ≈each node has to cope for ≈every 'added node', then
  - data structures, no./size of messages, and computation times "explode" despite the added resources (CPU, memory, links per added node)
    - problem is countered, e.g., by means of hierarchies (see above), KISS principle (see below), many principles & approaches of distributed algorithms (e.g., forwarding of accumulated values only)
    - often, measured via modified complexity theory (Big-O notation for number/size of messages, etc.)

Technical Scalability

Resuming technical scalability from above

- different subgoals defined in literature, different overlapping terms & definitions
- many concepts developed over time, which usually serve several of these different goals to a varying degree
- here: attempt to summarize as three top-level goals:

1. openness: graceful “spatial” diversity
   a. components of various origins (re-)usable in various systems
   b. ***and:*** systems (re-)usable in various environments
2. manageability under growth
   a. concentration: separation→ isolation→ concentration of concerns
   b. ***or*** deconcentration: autonomy of subsystems/components
3. longevity: graceful reaction to “temporal” change
   large systems bring about long life (cf. Internet) → make “robust”
   a. plasticity (ability to adapt to changing conditions, evolve)
   b. ***and:*** durability (ability to survive failure & “attack”)
   both can be approached via “proven properties” ***or*** “redundancy”

Economical Scalability

(Resuming economical scalability from above)

At first sight: not a computer science issue, but

- as business grows, so does IT infrastructure
- this requires mathematical and technical scalability of course
- in addition: the growing IT infrastructure, in its models / capabilities…, must reflect & support the growing business
  - in particular, a "vast variety of variants" (as business moves to other countries / regions → cultures/languages/legislations, but also to other market segments / customer "classes" /…)
- this leads to requirements almost identical to those from "technical scalability", but at a higher (application domain / biz) layer:
  - openness: interact with components/systems/environments "encountered" as business moves, diverges, & scales up
  - manageability … in spite of the many "variants"
  - longevity:
    - plasticity: adaptation to changing legislations/demands/…
    - durability: 24/7 support etc.

Scalability: Summary

Summarizing from above, we define

> *mathematical scalability* :==
> the ability of a solution to perform well under problem growth

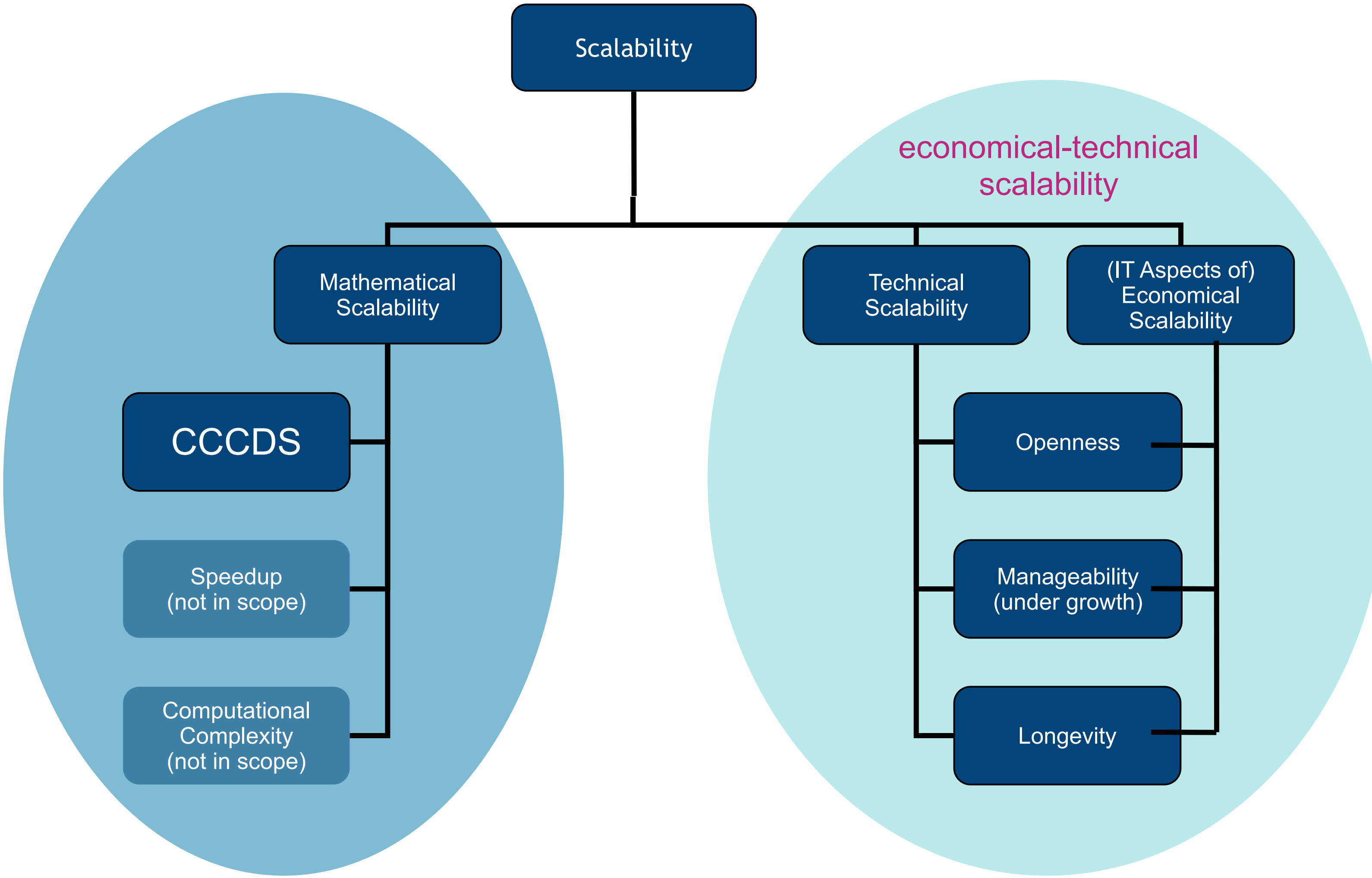
- solution may be a system, method, algorithm
- problem growth may refer to amounts of users, nodes, data
- "coping well" refers to performance

Economic scalability, as treated here (i.e. the IT aspect), often builds on technical scalability → in the remainder, we define

> *economical-technical scalability* :==
> the ability of a solution to cope well with spread

- concerns time (longevity) & space (geographic distribution, #of components)
- economically, spread may be over various contexts, countries, markets
- implies quest for radical reduction of human intervention

Scalability: Summary

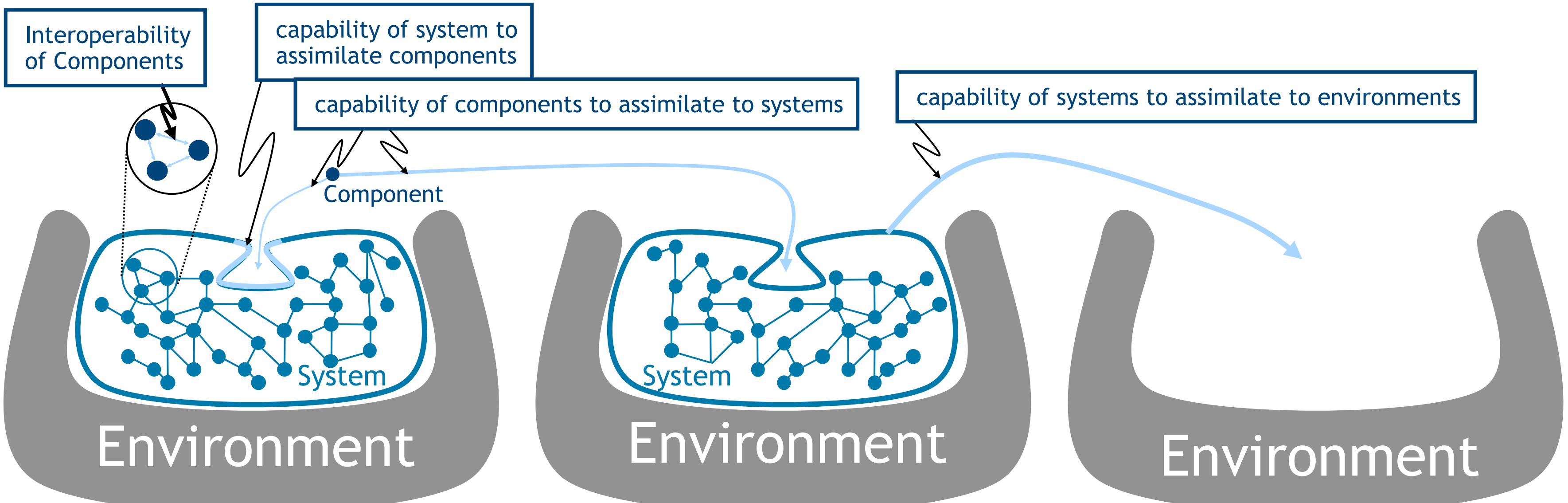


Openness: Standardization

UC requires "Openness" to advance from *Standards* (common) to Meta-Standards & Ontologies

- openness: 3 key types of entities must interoperate: components, systems, environments
- all 3 entities (usually): offer operations, exchange data (messages) → standardize data&operations
- Interoperation (hence, standards) concerns, with increasing complexity:
  - *syntax i.e. structure* of data, operations
  - *dynamics* (sequencing) of data, operations: important part of the
  - … *semantics* of data, operations: what is their effect?
- standardize syntax/dynamics/semantics or means (lges., calculi, …) for definition these (meta level)

Examples: communication protocols & services, software 'components' (ADT→ objects → components → services), ontologies, Peer2Peer/Overlay Networks, event based (pub/sub) middleware ...



Books

Primary Reference, available as PDFs (Details see Web Site):

Mühlhäuser, Gurevych: Ubiquitous Computing Technology for Real Time Enterprises, IGI 2007.

Additional Primary References for Chapter Connectivity (Mobile Computing):

1. J. Roth: Mobile Computing - Grundlagen, Technik, Konzepte. 2. Aufl., dpunkt 2005
2. F. Adelstein, S. Gupta et al.: Fundamentals of Mobile & Pervasive Computing McGraw Hill 2004,
3. Stefan Poslad: Ubiquitous Computing, Wiley 2009, ISBN 978-0-470-03560-3

SECONDARY Books:

1. J. Schiller: Mobilkommunikation, Pearson Studium 2003 (engl.: Mobile Communications, 2. Ed.)
2. D. A. Norman: The Invisible Computer, MIT Press Cambridge 1998
3. B. Walke: Mobilfunknetze und ihre Protokolle 1, 2, Teubner Stgt. 2000
4. K. Kelly: Out of Control, Perseus Books, Reading, MA; USA, 1994
5. F. Stajano: Security for Ubiquitous Computing, Wiley 2002
6. U. Hansmann: Pervasive Computing Handbook. The Mobile World, Springer 2003
7. J. L. Encarnaco: True Visions. The Emergence of Ambient Intelligence, Springer 2006
8. W. Weber, J. Rabaey, E. Aarts: Ambient Intelligence. Springer 2005
9. F. Gillert: RFID für die Optimierung von Geschäftsprozessen, Hanser 2007
10. Grigoris Antoniou: A Semantic Web Primer, MIT Press 2004
11. E. Fleisch, F. Mattern: Das Internet der Dinge, Springer 2005
12. J. Roth: Mobile Computing - Grundlagen, Technik, Konzepte dpunkt 2005
13. M. McTear: Spoken Dialogue Technology, Springer 2004
14. M. Dahm: Grundlagen der Mensch-Computer-Interaktion, Pearson Studium 2006
15. D. Jurafsky: Speech und Language Processing, Prentice Hall 2000
16. A. Greenfield: *Everyware : The Dawning Age of Ubiquitous Computing*, New Riders 2005
17. Conference Proceedings of: Intl. Conf. Ubiquitous Computing (Ubicomp) , Intl. Conf. Pervasive Computing (Pervasive) , IEEE Intl. Conf. Pervasive Computing & Communications (Percom) , Intl. Symp. Pervasive Computing & Applications (SPCA)