
Sisteme distribuite – Tehnologii

7. Ubiquitous* computing, UbiComp: calcul omniprezent

yoo-**bik**-wi-tuh s, omnipresent

Definitie Ubicomp

- Termen introdus de Mark Weiser in 1991
 - Viziune: o lume a calculatoarelor omniprezente care devin invizibile fiind incastrate in mediul fizic cu scopul de a ajuta oamenii ne-obstructiv in realizarea sarcinilor lor
 - Numit si ***pervasive computing*** – calcul pervasiv
 - Exemple pentru aplicatii ubicomp: un sistem de alocare a camerelor *inteligent*
 - Scaunele sunt capabile sa simta starea ocuparii lor si utilizeaza aceasta informatie pentru a deriva automat nivelul de ocupare a incaperii, care este afisat pe o placa electronica a usii si in hol pe o tabela pentru orientare
 - Facilitati tehnologice:
 - Consista din dispozitive de calcul numeroase, specializate si fara fir incastrate in mediul fizic
 - Dispozitive care percep & controleaza anumiti parametrii a mediului lor fizic & care pot comunica intre ele
 - Dispozitivele utilizeaz modalitati ergonomice, intuitive si ne-obstructive pentru a interactiona cu oamenii
-

De ce acum si ce?

- Avans tehnologic recent ce suporta ubicomp in
 1. procesoare,
 2. stocare,
 3. comunicare wireless,
 4. senzori,
 5. aprovizionare cu energie,
 - Construirea de tehnologii pentru energie care sunt eficiente este de importanta majora (dispozitivele pentru ubicomp sunt adesea wireless).
 6. dezvoltarea de materiale noi
 - materiale noi (ex. Displayuri flexibile, baterii minuscule) ce permit construirea dispozitivelor cu forme neconventionale
 - Tendinta catre “mai mult, mai destept, mai ieftin, mai putina energie” va permite construirea de sisteme ubicomp ale viitorului
 - O perspectiva din punct de vedere al informaticianului:
 - algoritmi, protocoluri & arhitecturi noi sunt necesare pentru
 - administra & controla cantitatea enorma de dispozitive de calcul legate in retea &
 - a gasi sensul pentru cantitatea imensa de date colectate de dispozitivele echipate cu senzori
-

Provocari ale middleware-ului

- Resurse cu constrangeri
 - Dinamica rețelei
 - Scara lansării
 - Integrarea în lumea reală
 - Colectarea, procesarea și stocarea datelor de la senzori
 - Integrarea cu infrastructurile existente
-

Resurse cu constrangeri (1/2)

- Pentru a permite integrarea ne-obstructionata in obiecte fizice si mediu -> dispozitivele adesea trebuie sa fie fara fir si sa satisfaca constrangeri de marime
- Marime si energie limitate => resursele precum puterea de calcul, marimea memoriei, latimea de banda pentru comunicare si distanta de transmitere sunt limitate
- Exemplu: un dispozitiv cu senzori de marimea unei cutii de chibrituri dezvoltat la UC Berkeley, MICA mote
 - ❑ Procesor 8-bit cu 8 MIPS,
 - ❑ 8 kilobytes RAM,
 - ❑ 128 kilobytes program memory
 - ❑ Communication bandwidth: 40 kilobits per second
 - ❑ Range: up to 30 meters
 - ❑ Ruleaza saptamani sau luni folosind o pereche de baterii AA.

MICA
WIRELESS MEASUREMENT SYSTEM

- ▼ 2nd Generation, Tiny, Wireless Smart Sensors
- ▼ TinyOS - Unprecedented Communications and Processing
- ▼ AA/Year Battery Life
- ▼ Small Form Factor
- ▼ Wireless Communications
- ▼ Light, Temperature, Acceleration/Seismic, Acoustic, and Magnetic Sensors
- ▼ Developed for DARPA NEST Program

Applications

- ▼ Wireless Sensor Networks
- ▼ Security, Surveillance, and Force Protection
- ▼ Environmental Monitoring
- ▼ Large Scale Wireless Networks (1000+ points)
- ▼ Distributed Computing Platform

Processor and Radio Platform (MPR300CB):
The MPR300CB is based on the Atmel ATmega 128L. The ATmega 128L is a low power microcontroller which runs TOS from its internal flash memory. The 128L has been selected for its low power and other features. The MPR300 (MICA) uses an ISM band radio transceiver module for wireless communication.

Sensor Boards:
Various sensor boards are available from Crossbow and other sources. These boards connect onto the MICA through a surface mount 51-pin connector. The 51-pin connector supports Analog Inputs, I2C, SPI, UART, and a multiplexed Address/Data bus. These interfaces make it easy to connect to a wide variety of external peripherals. Crossbow supplies the following sensor boards:

- MTS101CA Photo diode/Thermistor/Proto and Experiment Board
- MTS300CA Photo diode, Thermistor, Microphone, and Sounder
- MTS310CA Same as MTS300CA but also including Magnetic and Acceleration Sensor

The MICA mote is a second generation mote module used for research and development of low power, wireless, sensor networks. The MICA mote was developed by UC Berkeley's research group on wireless sensors. It consists of:

- Plug-in sensor boards
- TinyOS (TOS) Distributed Software Operating System.
- Atmega 128L processor
- 916MHz or 433MHz transceiver
- Attached AA(2) battery pack.

TinyOS is a small, open-source, energy efficient, software operating system developed by UC Berkeley which supports large scale, self-configuring sensor networks. The source code and software development tools are publicly available at: <http://webs.cs.berkeley.edu/tos>

TOS includes:

- Radio messaging
- Message hopping from mote to mote
- Low power modes
- Sensor measurements and signal processing

MPR300CA Block Diagram

80 Document Part Number: 6020-0061-01

crossbow technology, inc • 41 daggett drive • san jose, ca 95134-2109

Resurse cu constrangeri (2/2)

- Resursele limitate trebuie sa fie partajate intre aplicatii variate care sunt executate in retea si se serviciile de middleware insusi => serviciile de middleware ubicomp trebuie sa aiba o greutate redusa pentru a incapea in resursa
 - Middleware-ul ubicomp trebuie sa ofere mecanisme care sa ofere mecanisme pentru a ajuta la minimizarea cantitatii de resurse care sunt necesare pentru a indeplini anumite sarcini ale aplicatiei
 - Abordare: adaptare dinamica la performantele hardware, algoritmi, si protocoluri cf. cerintelor variate ale aplicatiei
 - Exemple:
 - Algoritmi adaptivi de fidelitate pentru utilizarea resurselor
 - Exploatarea cunostintelor despre aplicatii pentru a decide intreprerea din motive de eficienta a energiei
-

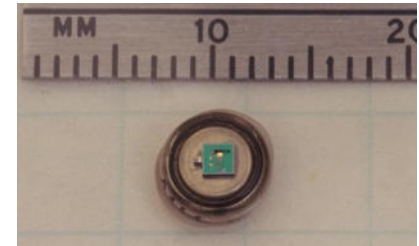
Dinamica rețelei

- Aplicatia Ubicom necesita colaborarea dispozitivelor distribuite spatial
 1. Dispozitivele in aplic. ubicomp tind sa fie inalt specializate, ex.
 - Masurare parametrii de mediu,
 - Extragerea informatiie de la datele colectate de la senzori
 - Interactiune cu utilizatori umani
 2. Numeroase aplicatii necesita input de la mai multe dispozitive spatial distribuite (ex. Nivelul de ocupare a unui incaperi).
 3. Resursele cu constarngeri ale dispozitivelor individuale adesea necesita colaborarea pentru rezolvarea unor sarcini complexe
 - Distanța de comunicare a dispozitivelor ubicomp fiind limitată =>
 1. Nu se aseamana cu rețelele de telefonie in care dispozitivele comunica direct c statia de baza
 2. Dispozitivele ubicomp formeaza **rețele ad hoc**:
 - Dispozitivele actioneaza ca rutere, forwardind mesajele de le vecinii lor peste multipl hopuri.
 - Dispozitivele mai puternice pot actiona ca si gateway care conecteaza parti ale rețelei ad hoc la o I infrastructura existenta
 - Topologia este subiect al schimbarilor dese datorita mobilitatii dispozitivelor sau restrictiilor mediului ce conduc la
 - Esecuri in comunicare (ex., un camion care trece printre raza de actiune a dispozitivelor),
 - Esecuri hardware (ex., baterii terminate, calare pe un dispozitiv).
-

Tratarea dinamicității rețelei

- *Ingrădire de informație*: download data în fazele online care vor fi posibil necesare ulterior
 - *Comunicare centrată pe date*: componente distribuite sunt identificate numai pe baza funcției pe care o îndeplinesc
 - ex., “anumite dispozitive în vecinătatea mea care pot măsura temperatura”.
 - avantaj: tolerarea faptului că dispozitivele pot trece offline prin schimbarea transparentă la un alt dispozitiv cu funcționalitate echivalentă
 - *Conceptul de închiriere*: resursele sunt alocate prin asociere cu un termen de închiriere care trebuie reînnoit în mod regulat
 - Dacă închirierea expiră datorită lipsei unei reînnoiri, sistemul poate în mod automat să reclame resursele asociate
-

Scara lansarii



- Exemplu: Smart Dust – concept introdus in 2001
 - Milioane de dispozitive de dimensiunea unei graunte de praf (motes) sunt lansate in mediu pentru a monitoriza diferite fenomene de mediu.
 - Un singur dispozitiv consta din senzori, un procesor, comunicatie wireless, sursa de energie
 - Dispozitivele sunt suficiente de mici, ex.
 - Stau suspendate in aer, de exemplu, pentru a monitoriza fenomenele de vreme sau calitatea aerului
 - Mixate in vopseaua de pe cladiri ca sa permita monitorizarea efectelor activitatii seismice asupra integritatii structurale ale cladirilor
 - Imposibil de configurare manuala, mentinere, fixare sau actualizare individuala a dispozitivelor datorita numarului foarte mare
 - Imposibil de asignat identificatori unici (ex., similar cu adresa unica MAC) pentru noduri individuale datorita surplusului de productie implicat
 - Situatie total simetrica (toate dispozitivele sunt initial identice), colectia de dispozitive trebuie *sa se configureze singura* pentru a atinge o stare operationala (ex., setarea topologiei retelei, asignarea sarcinilor la dispozitive, imbinare colaborativa si evaluarea datelor colectate).
 - Reteaua trebuie *sa se intretina singura* pentru a suporta esecurile nodurilor fara interventie manuala
- => Middleware-ul ubicomp trebuie sa ofere mecanisme pentru auto-configurare si auto-intretinere

Integrarea in lumea reala

- Prin definitie, dispozitivele ubicomp sunt incastrate in mediu fizic, in mod tipic pentru capturarea datelor despre mediu utilizand senzori atasati
 - ⇒ Ceea ce inseamna o incadrare a sistemelor ubicomp in lumea reala
 - ⇒ Timpul si locatie joaca un rol crucial in ubicomp.
 1. Este adesea important sa se cunoasca unde si cand se intampla ceva
 2. Timpul si locatia sunt cruciale pt. corelarea informatiei de la surse diferite

 - Dispozitivele ubicomp trebuie sa partajeze o intelegere comuna pentru timp si locatie cu scopul de a anunta daca sunt in aceeasi locatie si in acelasi moment de timp
 - Intelegerea comuna a timpului presupune sincronizare
 - Intelegerea comuna a locatiei presupune localizarea dispozitivului
 - ⇒ Necesare servicii care administreaza datele spatio-temporare
 - Ex. Un serviciu de locatie mentine o vedere actualizata a locatiilor curente ale dispozitivelor din retea
 - Ex. Un servicii al istoriei stocheaza locatia si timpul evenimentelor anterioare, oferind fundamentul pentru iterogari de genul “unde dispozitivele X si Y s-au intalnit in ultimul timp?”
-

Colectarea, procesarea & stocarea datelor de la senzori

- Functionalitatea de baza pentru ubicomp
- Senzorii colecteaza date de nivel jos (ex., serii de timp ale citirilor temperaturii)
- Aplicatiile sunt adesea interesate in facilitati de nivel inalt, precum “in conferinta” utilizat pentru a inchide in mod automat telefoanele mobile
 - Ex. functionalitate: *context service*:
 - Derivare a informatiei contextuale care cere evaluarea datelor de la senzori de diferite tipuri (ex., nivel de zgomot, intensitate lumina, calitate aer) care provine de la surse multiple
- Necesara pentru eficienta energiei si rationalizarea consumului de energie la comunicarea wireless
 - O mare cantitate de date brute de la senzori sunt transmise la o locatie centrala pentru procesare? Adesea nu este fezabil !!! Datorita consumului mare de energie, limitatilor de banda si probleme de scalabilitate
 - Solutie: “agregarea datelor in retea”
 - Datele de la senzori sunt preprocesate cat mai aproape de sursa pentru a reduce cantitatea de date care este transmisa
 - => reduce comunicatiile si salveaza energia pentru transmiterea agregatului compact in loc de date brute
 - => estompeaza separarea clara intre comunicarea si procesarea datelor care este tipica in sistemele distribuite traditionale si middleware-ul respectiv

Integrarea cu infrastructurile existente

- Tipic: dispozitivele ubicomp formeaza retele fara infrastructura de baza si ad hoc
 - Probabil: anumite dispozitive vor fi conectate la o infrastructura existenta precum Internetul.
=> Conceptul global de “Internet al lucrurilor” (ex. Programul EC FP7-ICT) conectand artefacte mici din intreaga lume.
 - Ratiuni pentru o asemenea integrare:
 - Infrastructura existenta poate fi folosita pentru diseminarea informatiei (ex. Statusul de ocupare a camerei) la destinatii la distanta
 - Infrastructura existenta poate oferi resurse (ex., putere de calcul, stocare) care nu este disponibila la dispozitivele tipice ubicomp.
-

Studiu de caz: rețele de senzori

- Subarie a ubicomp: “wireless sensor networks” (WSN).
 - WSN consta din *noduri senzori* — dispozitive autonome mici echipate cu senzori, facilitati de comunicare wireless, un procesor si o sursa de energie
 - Example: such a sensor node is the MICA sensing device
 - Contra-exemple (UbiComp, nu WSN): rețelele de dispozitive mobile
 - Aplicatii:
 - Biologisti: monitorizarea comportarii animalelor in habitatul lor natural
 - Cercetare a mediului: monitorizarea poluarii mediului.
 - Agricultura: observa calitatea solului si alti parametrii care influenteaza cresterea plantelor
 - Geologisti: monitorizeaza activitatea seismica si influentele sale asupra integritatii structurale a cladirilor
 - Militari: monitorizarea activitatilor in arii innacesibile
-

Model tipic de utilizare a rețelei de senzori

1. Un utilizator specifica o sarcina pentru senzori de nivel inalt -- ex., “Raporteaza salile in care nivelul de zgomot mediu depaseste un anumit prag”.
 2. Aceasta sarcina este impartita in mai multe sub-sarcini simple, fiecare fiind distribuita la noduri individuale din retea
 3. Aceste sub-sarcini colecteaza si pre-proceseaza citirile senzorilor de nivel jos
 4. Datele de la senzori sunt agregate si procesate pentru a forma un rezultat al senzorilor de nivel inalt care este raportat la utilizator
- Exista o cerinta impetuoasa pentru abstractizari care permit expedierea usoara a sarcinilor in retea ca intreg
 - Middleware-ul pentru rețelele de senzori trebuie sa suporte aceste abstractizari in programare
-

TinyDB

- Un numar de abordari trateaza retelele de senzori ca baze de date distribuite
 - Utilizatorii pot emite interogari de tip SQL pentru ca retea sa efectueze o anumita sarcina
 - TinyDB este un reprezentant pentru aceasta clasa
 - Suporta o baza de date singulara printr-un tabel “virtual”
 - Fiecare coloana corespunde la
 - Un tip specific de senzor (ex., temperatura, lumina) sau
 - Fiecare sursa de date de intrare (ex., identificatori de noduri senzori, puterea ramasa a bateriei).
 - Citirea la un nod poate fi privita ca o adaugarea unei linii
 - Limbajul de interogare este o submultime a SQL cu anumite extensii
-

TinyDB exemplu

- Numeroase camere sunt echipate cu senzori multipli
 - Fiecare nod este echipat cu senzori pentru a masura volumul acustic
 - Tabelul contine
 - Numarul camerei un care este senzorul
 - Etajul la care se afla camera
 - Volumul
 - ? Determina camerele de la etajul 6 unde volumul mediu depaseste pragul 10
 - R: Query:

```
SELECT AVG(volume), room // a pair of average volume and the respective room number is returned
FROM sensors
WHERE floor = 6 // selects rows from sensors at the 6th floor
GROUP BY room // selected rows are grouped by the room number
HAVING AVG(volume) > 10 // the average volume of each of the resulting groups is calculated
// only groups with an average volume above 10 are kept
EPOCH DURATION 30s // query is re-executed every 30 seconds
```
-

Abordarea TinyDB

- Utilizeaza o abordare descentralizata:
 - Fiecare nod are procesorul propriu de interogare care pre-proceseaza si agregheaza datele pentru a le trimite la utilizator
 - Executarea unei interogari implica urmasorii pasi:
 1. Un arbore de acoperire a retelei este construit si mentinut desi reseaua se schimba utilizand un control de tip inundare
 2. O interogare este difuzata la fiecare nod din retea de la radacina inspre frunze
 - O planificare este stabilita, astfel incat parintele si fiii sunt de acord asupra unui perioade de timp in care parintele va primi datele de la fii
 3. La inceputul fiecărei epoci, nodurile frunza obtin o copie din tabel prin citirea datelor de la senzor
 4. Frunza aplica criteriul de selectie.
 5. Daca criteriul este indeplinit, o stare partiala este creata continand toate datele necesare (ex, numarul camerei, numarul etajului etc).
 6. Starea partiala este expediată la parinte la timpul planificat.
 7. Parintele asteapta stările parțiale la fii
 8. Parintele procedeaza la fel, aplica criteriul de selectie si genereaza o stare partiala
 9. Starea partiala este trimisa la parintele parintelui
 10. Procesul continua pana la atingerea radacinii
 11. La radacina, starea este evaluata pentru a obtine rezultatul interogarii
 12. Întreaga procedură se repetă în fiecare epocă.
-

SensorWare

- Clasa de abordari inspirate din coduri mobile si agenti mobili:
 - Reteaua de senzori este insarcinata prin injectarea unui program in aceasta
 - Poate colecta date de la senzorii locali
 - Poate migra fara cu stare sau se poate auto-copia la alte noduri
 - Poate comunica cu copiii la distanta.
 - SensorWare este un reprezentant al acestei clase
 - Programele sunt specificate in Tcl:
 - Interogare: se considera un nume de senzor (ex., volum) si o comanda ca parametrii
 - Valoare: utilizata pentru a obtine o citire a senzorului.
 - Expediere: se considera o adresa si un mesaj ca parametrii si se expediaza mesajul la nodul specifica
 - Replicare: se considera una sau mai multe adrese ca parametrii si se trimit copio multiple ale scriptului la nodurile la distanta
 - Asteptare: set de nume de evenimente ca parametrii si suspendarea executiei scriptului pana cand apare unul dintre evenimentele specificate
 - Ocurenta unor evenimente asicrone (ex, receptionarea unui mesaj, expirarea unui timer) este reprezentata pentru fiecare de un eveniment.
-

DSWare

- O alta abordare pt. middleware bazate pe notiunea de evenimente.
 - Aplicatia specifica interesul in anumite stari a lumii reale (“evenimente de baza”).
 - La detectarea unui eveniment, un nod trimite o notificare a evenimentului catre aplicatiile interesate
 - Aplicatia poate de asemenea specifica anumite sabloane pentru evenimente (“evenimente compuse”), a.i. aplicatie este notificata pentru evenimentele care se potrivesc cu acest sablon

 - DSWare este un reprezentant al acestei clase.
 - Suporta specificarea si detectarea automata a evenimentelor compuse
 - O specificare de eveniment compus contine:
 1. Identificator de eveniment,
 2. Un domeniu pentru detectie specificand aria de interes,
 3. O durata a detectiei specificand perioada de timp de interes
 4. O multime de noduri senzori interesati in evenimentul compus
 5. O fereasra de timp W ,
 6. O functie de confidenta f ,
 7. O confidenta minima c_{min} , si
 8. O multime de evenimente de baza E .
 - Functia de confidenta mapeaza E la o valoare scalara.
 - Evenimentul compus este detectat si livrat la nodurile interesate, daca $f(E) \geq c_{min}$ si toate evenimentele de baza apar in fereastra de timp W .
-

DSWare exemplu

- Detectarea unui eveniment de explozie – necesita:
 - ocurența unei evenimente lumina (ex., lumina flash),
 - Un eveniment în temperatură (ex., temperatura ambiant ridicată),
 - Un eveniment de sunet (i.e., bang) într-o fereastră de timp de nivel subsecundelor W .
 - Funcția de încredință este definită:
 $f = 0.6 \cdot B(\text{temp}) + 0.3 \cdot B(\text{light}) + 0.3 \cdot B(\text{sound})$
(detectarea evenimentului de temperatură oferă cea mai mare încredință în explozie decât detectarea evenimentelor de sunet sau lumina)
 - Funcția B mapează un eveniment ID la
 - 1 dacă evenimentul respectiv a fost detectat în fereastra de timp W , și
 - 0 altfel.
 - Cu $c_{\min} = 0.9$, f va indica un eveniment de explozie dacă evenimentul de temperatură a fost detectat împreună cu cel de sunet și lumina.
-