Embedded Intelligent Systems

Dobrowiecki, Tadeusz
Lőrincz, András
Mészáros, Tamás
Pataki, Béla

Embedded Intelligent Systems

írta Dobrowiecki, Tadeusz, Lőrincz, András, Mészáros, Tamás, és Pataki, Béla

Publication date 2015

Tartalom

Embedded Intelligent Systems

1. Introduction

Embedded intelligent systems:

1. treatment from "embedded" perspective - already covered by embedded systems, sensor networks, automotive systems, real-time operating systems, cyber-physical systems, etc.

2. from "intelligent" perspective, i.e. from knowledge management and algorithmization angle.

Rationale:

The embedded technology does not support automatically the emergence of intelligence.

To have it one must carefully explore the algorithmization possibilities along the general AI lines, but taking into account the limitations and constraints of the technologies and typical applications.

Such limitations affect primarily not the implementation level, but the abstract knowledge management level leading to conceptually new and involved tasks:

- how to "implement intelligence",

- what aspects of intelligence feed well the "embedded" demands,

- how can intelligence utilize the "embedded" technologies to provide qualitatively better, new services to the human users.

The basic "embedding" means being hidden in some common "not exactly computer system" artifacts, like TV set, automobile, airplane, surgical robot, coffee machine, fridge, etc.

Only a small conceptual step further is needed, i.e. the realization that the whole human environment is full of such "embedded computer system devices" and as plenty of technologies of yesterday (e.g. wood, steel, electrical drives, etc.) disappeared already in the normal human environment, the informatics is rushing toward the same destiny, i.e. to disappear as a conceptually separate, outstanding technology, and to become rather a part of qualitatively new "usage of things" (even such large like a building or a ship).

Similarly to the hierarchical connection and relation between physical and technological systems, computer systems embedded into them and providing services to their users must appear at various abstraction levels and must be hierarchically related.

In this sense the intelligence behind the embedded systems, providing meaningful local functions, yet interconnected into "global" embedded systems, serves local aims, but is fused together into meaningful global services is that of so called "ambient intelligence", i.e. intelligence also "embedded" in the human environment.

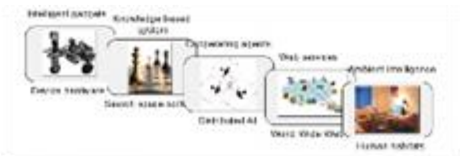
In a sense an intelligent embedded system cannot be intelligent in different way as being ambient intelligent, and vice versa an ambient intelligent system must be by its very nature be an embedded (however large and global) system.

2. 1 From traditional AI to Ambient Intelligence. Embedded and multiagent systems, pervasive computing techniques and ambient intelligence.

2.1. 1.1 The essence of Ambient Intelligence (AmI) paradigm

2.1.1. AmI - as the next step in the progress of the traditional AI

"pure" hardware neural networks, cybernetic gadgets computer MYCIN expert system, STRIPS representation, knowledge-based systems network agents, personal secretaries, mediating agents www ontology, browsers, recommending systems human habitat AmI, Embedded Intelligent Systems



1. ábra. The evolution of intelligent systems

2.1.2. Ambient Intelligence (ambient temperature - room temperature)

Latin ambiens, ambient-, ...ambire, to encircle, ...

Typical appearances:

Intelligent Room (plenty of research projects)

intelligent college, intelligent lecture room, intelligent nursery, intelligent surgery

(disaster response, SAR, ...) Monitoring and Information Center, ...

Intelligent Space

...workspace/ living space

school, nursery, classroom, office, lecture room, surgery, SAR center, military space, ...

living quarters, flat of a patient, flat of an elderly, hospital, hospital ward, ambulance, senior nursing home, car, ship, greenhouse, space station, deep see lab, college, Biosphere, ...

e.g.: iDorm Ambient Intelligent Agent Demonstration

http://www.youtube.com/watch?v=NtxkEYLeLZc

Smart House, Smart Home (industrial projects)

...

AAL - Ambient Assisted Living (serious EC support)

AAC - Ambient Assisted Cognition (serious EC support)

Characteristic paradigms:

Agent system= the source/provider of intelligence

its environment = the source of problems, source of information

other agents in the environment

multiagent systems - agents from environment perceived as entities

models, communication, cooperation, conflicts ...

(novelty) agent $\Longrightarrow$ an entity in the user environment (novelty) agent environment = user environment

user $\Longrightarrow$ entity in agent environment

user environment a component in the agent environment

Embedded (intelligent) system

Pervasive (ubiquitous) system, computer technology permeating all, everywhere, ...

challenge: user interaction with the vanishing computer technology

- physical disappearance: miniaturization = wearable, implanted, ...computer embedded, outward only visible is the function of the embedding
- mental disappearance: embedding: computer an invisible part of a visible artifact
- cognitive disappearance: no more a computer, but an information devicefor communication, for cooperatione.g. interactive wall (DynaWall 4.4 m x 1.1 m),e.g. interactive table (InteracTable)
- emotional disappearance: high emotional charge, attraction of the artifact

Ambient Intelligence

1999, Advisory Group to the European Community's Information Society

Technology Program (ISTAG) (establishing the notion)

2001, ISTAG 'Scenarios for Ambient Intelligence in 2010'

social challenges (social acceptance, always be in control,

AmI paradox: physically pervasive, vanishing device -

psychologically may be obtrusive)

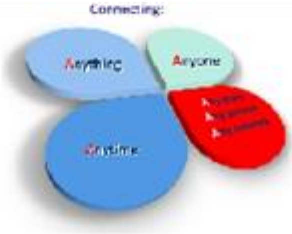
message: one must be in control in the AmI, non professional user also

technological challenges (miniaturization and networking,

advanced pattern recognition, safety)

Cyber-Physical Systems, CPS - Integration of computations, networking and physical processes

Internet of Things, IOT - "Smart things/objects": active components of information processes. Enabled to interact and communicate by exchanging data and information "sensed" about the environment. React autonomously to the "real/physical world" events and influence it by running processes that trigger actions and create services with or without direct human intervention.



2. ábra. **Internet of Things** (from http://www.internet-of-things-research.eu/about_iot.htm)

2.2. 1.2 Special characteristics of the Ambient Intelligent systems (SRATUI)

- sensitive (S): equipped with devices, technology (sensors, etc.) permitting to sense the state of the environment and its components,
- responsive (R): reacting (initiating computations) upon observation of changes in the environment,
- adaptive (A): the result of the reactive computations may influence the architecture of the further information processing and thus the character of the services provided by the system,
- transparent (T): with traceable causality in function from sensitivity to responsiveness and adaptivity (easy to comprehend in its actions by the user),
- ubiquitous (U), pervasive: hidden in the everyday non computational and traditional environment (e.g. as a smart fridge, intelligent garage, smart heating systems etc.)
- intelligent (I): exercising computationally implemented intelligence to provide the proper level of responsiveness and adaptivity,
- distributed in space: because the environment is spatially distributed, i.e. the intelligent computation must recognize it and turn to its advantage (spatial reasoning),
- temporarily permanent: because the state of the environment and its components can be meaningfully comprehended only if followed in time, the character of the applications is 7 days a week/ 24h a day, and the sensed information requires temporal fusion to be transformed into useful knowledge.

Challenges to AI = knowledge representation and knowledge manipulation

interpretation of the environmental states (based on what?)

representing environmental information and knowledge,

representing, modeling, and simulating environmental entities (based on what?)

designing actions and decisions (who/what is deciding and acting?)

learning the environment (but there are also people acting independently in the environment!)

interaction with people

interaction in the environment, interaction with the environment,

most frequent interaction medium is the (natural) language,

the richest sensory input is usually a vision

affecting, restructuring the environment, ... (but what happens, if the humans counteract via available actuators and devices?)

3. 2 Review of ambient intelligent applications: smart homes, intelligent spaces (Ambient Assisted Living, Ambient Assisted Cognition)

3.1. 2.1 Special and important AmI applications

- Smart House, Smart Home, Intelligent Home, Cooperative Building, ...

3.1.1. Smart House (Phillips taxonomy)

- controllable house
  - house with integrated remote control (e.g. integrated VCR and TV RC)
  - house with interconnected devices (e.g. wifi connection between TV and video recorderhouse controlled with voice, gesticulation, and movement)
- programmable house
  - programmed for timer or sensor input
  - thermostat, light controlled heating, light control, ...
  - house programmed to recognize predefined scenarios
- intelligent house
  - programmed for timer or sensor input
  - house able to recognize behavioral schemes

What a smart house can do

- communication/ phone control
  - house inward - settings, commands
  - house outward - announcements, redirecting, (e.g. fence, door phone)
- integrated safety
  - holistic management of alarms, safety increasing house behavior (lights, sounds, simulation of human presence)
  - sensory perception + autonomous action (e.g. smoke sensor - calling fire department, controlling lights, opening doors, loud speaker announcements, ...)
  - safety - Physical Access Control and Burglary Alarm systems
  - safety - health and well being of the inhabitants (prevention, monitoring)
  - safety - safe design (and materials), monitoring and controlling the "health" of the building,
- electrical-mechanical locks and openings, magnetic cards, RFID stamps, biometrics house automation (basic household "life-keeping" functions), maintenance
  - independent life style integrating house control and other devices for independent living (e.g. wheelchair, elevated cupboards, sinks, ..., proper light control for usual getting up at night, ...),
  - refreshment and hygiene teeth brushing, hear brushing, make-up in front of mirror in the mirror clock, the news, weather forecast, display of weight, blood pressure, ...
  - easier life (setting curtains, lights, hot water, bath, news, ..., watching TV, channel, voice level, taking sound level of other media down, lights, curtains, ...), office work at home

3.1.2. Cooperative Buildings

Concept introduced in 1998. Emphasis: the starting point of the design should be the real, architectural environment ('space' can sometimes be also 'virtual' and/or 'digital').

The building serves the purpose of cooperation and communication. It is also 'cooperative' towards its users, inhabitants, and visitors. It is 'smart' and able to adapt to changing situations and provide context-aware information and services.

Basic problem -

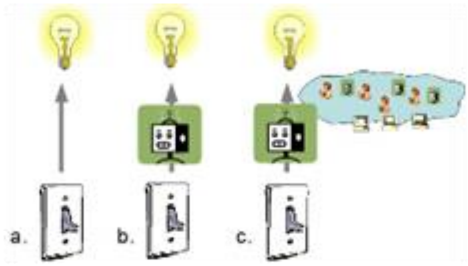
1. the house can be even smarter!
2. the industrial design is not drawing from AI results
3. AI research - only slowly realizes that this is an excellent testand application area, with interesting and serious benchmark problems.

3.2. 2.2 Intelligent Spaces

3.2.1. Adding intelligence to the smart house environment

domotica (intelligent household devices)

- switch - x - lamp
- switch - local intelligence (agent) - lamp
- switch - global intelligence (agent system) - lamp



3. ábra. **Domotica:** (a.) traditional manual device, (b.) embedded system supported by local intelligence, (c.) intelligent device drawing support form the global (physical and virtual) ambient environment.

Creating an Ambient-Intelligence Environment Using Embedded Agents,

Nov/Dec 2004, 19(6), 12-20,

http://www.computer.org/portal/web/csdl/doi/10.1109/MIS.2004.61

Inhabited intelligent environments,

http://cswww.essex.ac.uk/Research/intelligent-buildings/papers/2203paper22.pdf

3.2.2. Why this problem is difficult?

Smart House (or similar):

- environment: from the AI point of view the environment of a smart house (or similar applicative environment) constitutes the most difficult environment regarding the design and the management of intelligent systems.

This environment is:

- not accessible - i.e. not every (even essential) information can be obtained via sensors, due to the technological (nosuitable technology exists) and/or implementation problems (too complicated, too expensive)
- dynamic - the environment is in constant change due to the activities of the inhabitants and the change of the externalphysical environment (e.g. day/night cycle)
- stochastic - the causal chains may be so complicated that they are impossible to be modeled deterministically, stochastic models must be used (which by definition are screening the details
- not episodic - the activities are going on the 7 days a week/ 24h a day basis, demanding similar continuous computational activity from the monitoring system
- continuous - the environment is basically continuous in space and time, every discretization means loss of detail...

consequently the knowledge is always missing and uncertain

human agent (inhabitant) in the Smart House

- uses space towards his/her own (not known) goals
- it is a user which moves in the space
- it is a user which changes with time
- it is a non professional user, basically used to inter human interactions, but not to human-computer interactions
- s/he may be degraded in his/her faculties (a child, an elderly, ...)
- interactions, movements, goals are affected by the physical, mental, and emotional state of the user, ...(not a usual human-computer interaction)this state must be perceived
  - context-dependent computer techniques
  - affective computer techniques
- mixed human-agent/ robot/ softbot teams present in the space
- defending the privacy - privacy-sensitive computer techniquesQuality of Privacy (QoP) (available technology cannot be used fully) qualitative feelings of the users aspects: location, identity, activity, access, ...

HCI - typology of the interactions/ interfaces

- HCI (direct)
  - traditional (...keyboard)
  - artifact management (...joystick)
  - natural interfaces (speech, sound, and picture/video)
  - modalities
    - controlled natural languages
    - natural language based device interfaces

    - emotional interfaces

- HCII (intelligent HCI)
- iHCI (Implicit Human Computer Interaction) (sensory observation of the user)interaction of the human and the environment (devices) toward a single goal, within it implicit input from the userimplicit output toward the user, connection via the context
  - Implicit Input: such human activities, behaviors, which happen for the sake to reach some goal and do not mean necessarily a contact with the computer system. It will become an input if the system will recognize it as such and interpret properly.
  - Implicit Output: Such computer output which is not an explicit computer system output, but a natural component of the environmental and task dependent user interactions.The essence: the user is focusing on the task and is in contact rather with his/her physical environment and not with some explicit computer system.

What is an implicit Input/ Output good for?

- proactive applications, triggers, controlknowledge of events, knowledge of situations
  - application triggering (start, stop, tip. in warning, alarm systems)
  - choice of application depending on the situation
  - passing the actual situation as a parameter to an application (e. g. navigation)
- adaptive User Interfaceuser interface adapting to the situation
  - traditional: conditions, circumstances of the usage known
    - design: interface fitting the situation optimally
  - situation dependent:speeding up, simplifying the presentation of information in case of dangerin case of busy user - choosing modality least affecting his/her activitysafeguarding privacy in a given situation
- communicationsituation = filtering communication
- resource management

3.3. 2.3 Components of intelligent environments

3.3.1. Physical space, physical reality

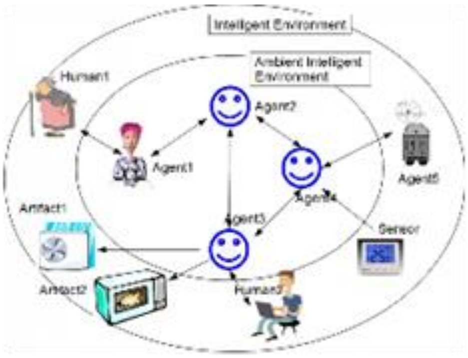
- human (and/or other animal, botanical) agents
- physically present robotic agents (e.g. vacuum cleaner)
- space "own" devices, for
  - interactions human $\longleftrightarrow$ human
  - interactions human $\longleftrightarrow$ physical space (life conduct, work, ...)
- effectors, actuators AmI $\longrightarrow$ physical space
- sensors AmI $\longleftarrow$ physical space
- communication interfaces AmI $\longleftrightarrow$ humans

3.3.2. Virtual space, virtual reality

- (virtual) agentsroles, organizations, communication, agent mental modeling, ..."visible" and "non-visible" agents
- space "own" devices, for
  - interactions agents $\longleftrightarrow$ agents
  - interactions agents $\longleftrightarrow$ virtual space
- effectors agents $\longrightarrow$ virtual space
- sensors agents $\longleftarrow$ virtual space
- communication interfaces agents $\longleftrightarrow$ human agents $\longleftrightarrow$ agents AmI $\longleftrightarrow$ agents
- agent - agent interfaces (agents may be also human):one $\longleftrightarrow$ one, done one $\longleftrightarrow$ many, easily done many $\longleftrightarrow$ one, difficult many $\longleftrightarrow$ many yet not existing (technology, protocols)

Agents Visualization in Intelligent Environments,

http://research.mercubuana.ac.id/proceeding/MoMM462004.pdf



4. ábra. **Physical vs. virtual environment:** agents in the ambient intelligence environment are in contact with sensors, other agents, and human inhabitants. Human inhabitants are primarily in contact with each other and the environmental non-computational artifacts.

Spectrum of possible "realities":

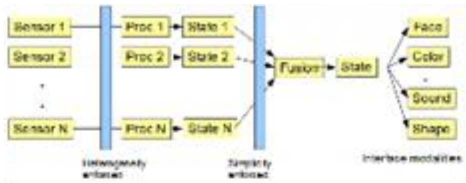
- virtual reality human is sensing entities embedded in virtual reality, ...
- augmented ("helped") reality human is obtaining sensory information belonging to multiple senses or activities focused in a single modality (e.g. a modern pilot helmet with multiple dials projected visually into the visual screen of the helmet)
- hyper-reality the usual phenomena obtain (are "enriched" with) attributes nonexistent in the "normal" reality, these however are sensed by the human with his/her normal senses (modalities) (e.g. water flowing from the faucet is enlighten with a color (blue-red) reflecting its temperature)

3.3.3. Sensors

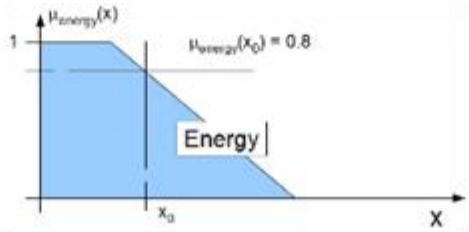
- suitable diversity
- strategic placement (e.g. movement sensors)

- type of the sensory data
  - movement sensors
  - repeated body movement sensors
- interaction between the inhabitants and the intelligent space objects
  - (refrigerator, window, door, medicine container, ...)
  - sensors on important objects sensing the change of state
- sensor networks communication, sharing information, energy management, intelligent sensor network
- SensorWebOGC Sensor Web Enablement Standards, SensorML OGC Sensor Web Enablement: Overview And High Level Architecture, http://portal.opengeospatial.org/files/?artifact_id=25562
- sensor fusion (Bayes, Dempster-Shafer, fuzzy, Kálmán, ...)

3.3.4. E.g. Tracing the state of an AmI system with fuzzy logic



5. ábra. Flow of information in AmI applications. Information about different objects and by different sensory mechanisms is processed and transformed into a uniform internal representation. Various data are now fused (integrated and abstracted) into a joint description of the environment. The last step is to the decision upon the presentation, how to show these results to the user taking into account his/her perceptional capabilities and the available devices.



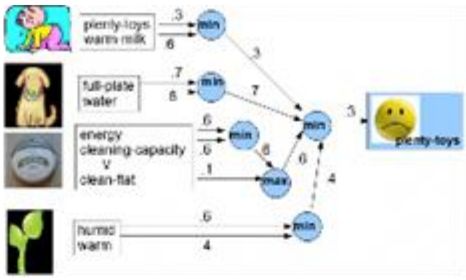
6. ábra. Inhabitants of an intelligent environment targeted to keep ,,happy". Then the environment is also ,,ok" and does not require corrective action. If the environment on the whole does not fares well, then the task is to identify which inhabitant is offended the most and to correct its situation by suitable actuator and action.

IE-comfortable = vacuum-cleaner-comfortable ∧ plant-comfortable ∧ dog-comfortable

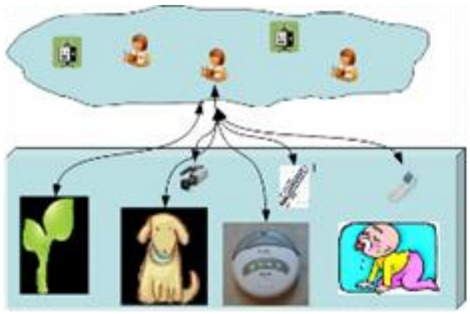
∧ child-comfortable = (energy ∧ cleaning-capacity) V

clean-flat) ∧ (humid ∧ warm) ∧ (full-plate ∧ water) ∧ (plenty-toys

∧ warm-milk)



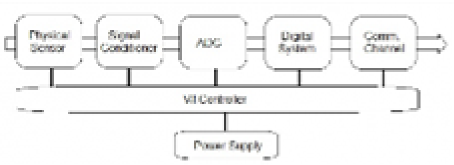
7. ábra. State variables can be expressed in fuzzy logic. That way the mathematical prediction of corrective actions can be implemented with simple computation (needing at the most a learning to tune the membership functions). Here a fuzzy set „(Having) Energy" is presented. Its universe is related to the course of time and its membership function emphasizes lower (early in time) values. In time-point x_0 the energy level was still ca. 80%



8. ábra. Monitoring state variables to judge which component (agent) suffers the most. The low value of the environment satisfaction can be traced to the non satisfactory state of the child.

Based on: Using Fuzzy Logic to Monitor the State of an Ubiquitous Robotic System, http://aass.oru.se/peis/Papers/jus08.pdf

3.3.5. Spaces and devices - Sensor design



9. ábra. Components of an intelligent sensor. (reference)

Low level information provider for context computations:

gathering, integration, processing, utilization

sensor information → system state → decision → effecting

- Data low level ambient characteristics - elementary adaptation ... integrated information - functional, model based adaptation, empathic computation
- Design
    - dedicated (designed as and for sensor)

- serendipitous (ad hoc) - electronic mass gadgets, cable TV, mobile, ...
  - cheap, easily available platform to put out sensors, cheap communication(webcam - movement sensor, mobile - diagnostic station e.g. for asthma, ...)
- iHCI (unconscious, implicit, from interactions, ...)

- Location
  - static location - static (ambient) characteristics
  - static location - dynamic characteristics (state change of fixed location objects - window, door, ...) (tracking human/ objects - movement sensors - microphone, video camera)
  - dynamic location - dynamic characteristics (tracking human/ objects - wearable (ID) sensors, RFID, iButton, ...) (state change of moving human/ objects - medicine container, ...)
- Passive/ active tip. all passive (pull, push) active - panic button (sensing change of state, push) Emergency Medical Alert (EMA) button, wearable, wireless connection with the center, ... "scenario" button (command device, but in the same time (iHCI) a state/ emotion/ intend sensor)

3.3.6. Functions in intelligent spaces

sensed quantity	function, usage
stress, pressure	floor, door, bed, stairs
position, direction, distance, movement	safety, localization, tracking, fall detection
light, radiation, temperature	safety, localization, tracking, healthcare-safety, energy efficiency
solid-, fluid, gaseous materials	safety and healthcare, monitoring, pool-management, watering efficiency
iButton	identification of human, objects
sound	safety, sound volume, speech recognition
picture	safety, identification, context recognition
bio	identification of human

1. táblázat.

3.3.6.1. Bio (authentication, identification)

biometric sensors (unique measurable non-varying biological characteristics representative to an individual)

physiological biometrics - specific differences in characteristics identifiable

with 5 sense-organs

(sight: looks, hair, eye color, teeth, face features, ...,

sound: pitch,

smell,

taste: composition of saliva, DNA,

touch: fingerprint, handprint)

behavior biometrics - style of writing, rhythm of walking, speed of typing, ...

fingerprint-readers iris-scanners hand/finger-readers (hand structure, build-up, proportions, skin, ...) face recognition sound/speech recognition signature dynamics, keyboard dynamics vein system recognition (new) (extremely low FN, False Rejection 0.01%, FP, False Acceptance 0.0001%, Pacific region, Asia), Joseph Rice, 1983, Eastman Kodak, http://www.biometriccoe.gov/ (FBI Biometric Center of Excellence) DNA ear smell, body smell recognition (machine odometer, artificial nose, ...) 2D bar-code readers coded with biometric information

3.3.6.2. Emotion (sensors)

- emotion recognition (sound pattern, facial expression, mimics, ...)
- physiological detection of emotion (change of physiological state = source of the emotion) anger, fear, sadness - skin temperature happiness - dislike, surprise, fear - sadness - heart rate
- physiological detection of emotion dynamics (BVP Blood Volume Pressure, SC Skin Conductance, RESP Respiration Rate, SPRT Sequential Probability Ratio Test, MYO, ...)

3.3.6.3. Picture processing - identifying and localizing people in space

- triggering location based eventschoosing the best audio/ video device to replay messages directed to particular persons
- identifying and using preference model characteristic to a particular location and user (lights, setting sound levels, ...)
- identifying/ understanding behavior of a particular person to compute suitable system actions

Requirements:

managing human locations and identity (resolution e.g. 10cm,

tracking color histograms, ...)

suitable speed ($\gg$ 1 Hz)

multiple human pictured in the same time

managing machine representation of appearing, disappearing

humans (delete, generating)

processing pictures from multiple cameras (lateral cameras instead

of ceiling cameras)

24h working regime

tolerance: partial occlusions and pose variations (Kálmán-filters,

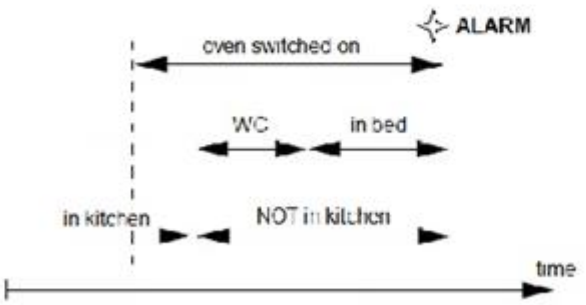
particle filters, ...)

3.4. 2.4 Knowledge intensive information processing in intelligent spaces

3.4.1. Reasoning

- modeling (human, activity)

- spatial-temporal reasoning
  - environment: "human path" - there exist a definite goal "topsy-turvy"
  - sensing timeliness: interaction human - object - long broke, what next
  - question of temporal granularity



10. ábra. Warning generation based on state and event observations (reference)

- causal reasoning
- case based reasoning
- ontological reasoning
- planning

3.4.2. Activity/ plan/ intention/ goal, ...recognition and prediction

- giving up plans, non-observable actions, ...
- failed actions, partially ordered plans, ...
- actions done for more simultaneous goals, state of the world, ...
- multiple hypotheses, ...
- Probabilistic Hostile Agent Task Tracker (PHATT)

3.4.3. Dangerous situations

- identification
- returning the environment into its normal state
- notification of the user

3.4.4. Learning

data- (time series-) mining

AI planning

- warning the user what to do

- finishing actions instead of the user, if needed

3.4.5. Modeling

- modeling behavior of the user, with good basic model - anomaly detection, identifying the emerging and abnormal behavior

smart ... home lifestyle patterns (e.g. proper dining and sleeping) hospital medicine intake (e.g. proper medicine in proper doses) office resource utilization (e.g. acts and courtrooms) auto driver behavior (e.g. increasing safety, if falling asleep) classroom interaction between teacher and students (e.g. focusing camera

on the proper object) monitored street

behavior monitoring (e.g. number plates of the speedsters) ...

3.4.6. Context - context sensitive systems

- context recognition
  - modeling human user
    - one-more, identity, face
    - localization problems
    - emotional state
    - prediction of actions, plans, intentions
    - tracking
  - modeling space
    - state, autonomous processes, ...
- based on context knowledge
  - interpretation of communication
  - interpretation of other sensory information (e.g. interpreting actions)
  - (body language, gestures, e.g. a human is able to recognize when his partner is in a hurry)
- relevant context information:
  - verbal context (direct communication) role division between communicating partners aim of communication, aims of individuals
  - local environment (absolute, relative, type of environment)
  - social environment (e.g. organization, who is there?)
  - physical, chemical, biological, environment.

3.4.7. AmI scenarios

analysis

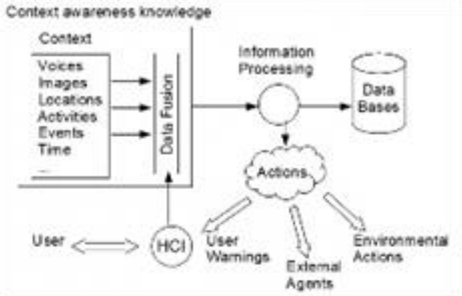
flow of information

controlling AmI

behavior of the technologies

3.4.8. Dark Scenarios (ISTAG) - future AmI applications - basic threats

monitoring users, spam, identity theft, malignant attacks, digital divide



11. ábra. Context awareness scheme.

3.5. 2.5 AmI and AAL - Ambient Assisted Living

Active ageing - World Health Organization - positive experience, continuous opportunities: health, participation, safety, increasing quality of life

active - continuous participation in social, economical, cultural, intellectual, societal affairs (not only solely

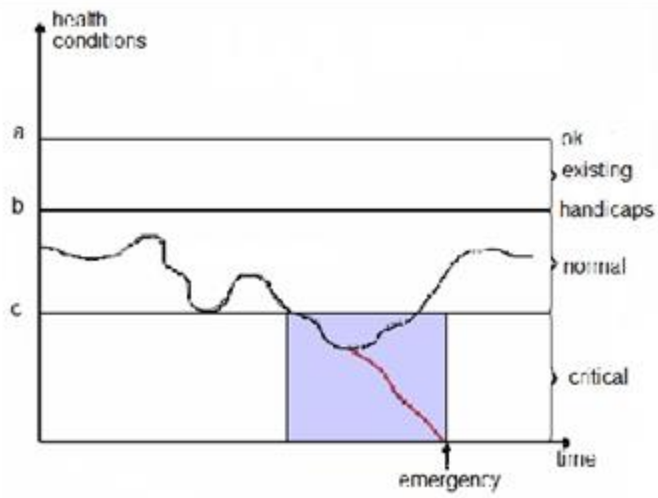
3.5.1. Basic services

1. management of emergencies
2. supporting autonomy
3. increasing of comfort

	management of emergence	supporting autonomy	increasing of comfort
„in-door”	emergency prediction, detection prevention	cooking, dining, drinking, cleaning, dressing, supporting taking medicines	logistic services, supporting finding things, supporting information gathering and recreation
„out-door”	emergency prediction, detection prevention	supporting shopping, supporting travelling, supporting bank transactions	supporting transportation, supporting orientation

2. táblázat.

3.5.2. Human disability model



12. ábra. Levels of human management. (after Living Assistance Systems - An Ambient Intelligence Approach, http://www.irisa.fr/lande/lande/icse-proceedings/icse/p43.pdf)

3.5.3. Robots and AmI

- helping elder people due to their physical limitations, managing household devices, during household activities (active command, delivery of objects, cleaning, ... passive, in ambient space, to decide what to do, e.g. pl. wiping, cleaning, ...)
- methodological components of robots - cognitive systems, affective reasoning(recognizing, interpreting, processing emotions, multi modal dialogues)
- walking/ robots walking people - getting up from the bed, bathing, ...- independent life without full time care-provider, keeping privacy intact
- assistive robots - therapy - adjustable level movement control, precision, patience to show or to expect exactly the same movements
- computerized "pets", healthcare services, company, socialization(measuring physiologically important signals, answer recording for questions, alarming, if long delay).

3.6. 2.6 Sensors for continuous monitoring of AAL/AAC well being

(cheap, reliable, not disturbing the user, fostering fusion, ...)

3.6.1. Activity of Daily Living (ADL)

- human with behavior difficult to predict - predictability needed for the general well being for the basic activities regular dining, sleep, using bathroom, washing/body care, taking medicines, dressing, walking, using phone, reading, cleaning, using bed, handling furniture, stairs, ...).
- predictability increases with age (where it counts the most - social environment of an elderly - more chance for a success)

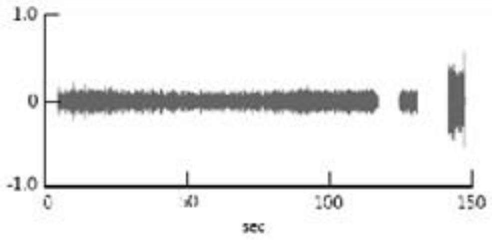
3.6.2. Special AAL, ADL sensors:

- tracking ADL cheaply and simply
- problematic ADL (critical, but complex sensor technology needed, low reliability)
  - taking/ handling medicines, handling money, ...

simply pressure sensors - using bed, using chair, presence/ room, opening flat doors, ... vibration or acoustic sensors placed on water pipes - water usage, ... video/IR cameras, ...

3.6.3. Monitoring water spending

- component of more important ADL
- industrial/ commercial solutions expensive and complicated (invasive, ultrasound, power supply, ...)
- no moving elements, for existing pipes, internal power supply, simple, cheap, ... acoustic signature of flowing water (short term) pipe temperature (long term)

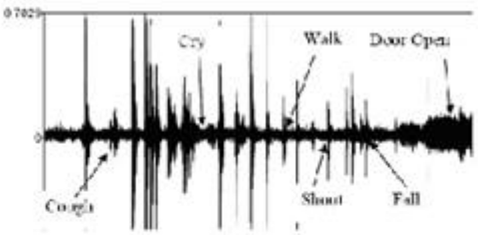


13. ábra. Acoustic signature of the water pipe (kitchen or bathroom) (from Sensor networks for continuous health monitoring, BT Technology Journal, Vol 22, No 3, July 2004)

Video-monitoring of vulnerable people at home environment

- detecting presence and localization: + 3D model of the furniture and the wall configuration
- robust identification: RFID technology more reliable than face recognition
- fall detection: real-time tracking of a patient, pose estimation (sitting, standing, laying, ...). alarm: laying, floor, ... not moving, in unusual place (temporal + spatial (logical) reasoning)
- activity monitoring: full day tracking, localization, statistics, "time spent in bed", "sitting in an armchair", "standing", ...
- keeping privacy, extend of usable "rough" information (here the picture)
- more processing close to sensor level, only context/ semantic information is sent to a remote storage for remote processing
- retrieving 2D silhouettes, detecting colors, fusion of multiple camera tracking

Audio-monitoring of vulnerable people at home environment



14. ábra. Sounds of the environment (after Audiovisual Sensing of Human Movements for Home-Care and Security in a Smart Environment, http://www.s2is.org/Issues/v1/n1/papers/paper14.pdf)

Social rhythms, routine behavior at night, daily activity, ...

basic behavior: time of going to bed, time of getting up, time spent in

bed,

number and duration of getting ups during the night, measures of

unrest, ...

Tactex BEDsensorTM pressure meter under the mattress, 24h

monitoring

sleep characteristics - actigraphy: ca. 1970, extending lab sleep measurements,

relating body movement and sleep cycle (characterizing patient, characterizing treatment, ...)

effectiveness of sleep (how much % of laying in bed is sleep)

sleeping latency (when falls asleep after getting to bed)

total sleep time

wake time after falling asleep -

number of waking ups during sleep time

number of getting ups during sleep time

3.7. 2.7 Sensor networks

- more sensor - better! (more reliable context, sensor fusion needed) ADL warning more reliable and precise
- in case of more ADL - context is such that the danger situations are difficult to recognize(context is not selective enough regarding danger situations)

E.g. reading, resting, ...easily covers serious health care problems

(in usual tract of time, patient is sitting in the favorite armchair not doing

anything, ...)

when he/she used to read - how to detect reliably the discomfort?

Deviation from the normal, which is difficult to observe for a long time

(if sitting for a long time, but then the warning comes late)

- solution: installing more sensors (panic button, scarcity of minute movements, ...),
- fusion with other sensors (e.g. covering the whole room: bed, carpet, legs of the chair, ...fusion of the pressure sensors, ...)

3.7.1. Possible sensor fusions

ADL \ Sensor domain	Water consumption	Electricity consumption	Noise/vibration	Gas consumption	Motion	Temperature	Mechanical movement	Pressure/weight	[illegible]	Location in room	Humidity
Toilet	•		•				•			•	
Bath/shower/washing/grooming	•		•			•				•	•
Bed/sleeping			•			•		•		•	
Chair/sitting			•			•		•		•	
Dressing							•			•	
Light exercising			•		•		•	•		•	
Walking			•		•			•		•	
Stairs/steps			•					•		•	
Reading/relaxing								•		•	
Watching television		•	•			•		•		•	
Telephone			•				•		•	•	
Opening front door			•			•	•			•	
Feeding (drinking), preparing hot food/drink	•	•	•	•		•	•	•		•	•
Household tasks — washing clothes, vacuuming, etc	•	•	•	•	•	•	•	•			•

15. ábra. Examples of sensor fusion bringing generation based on state and event observations (from Sensor networks for continuous health monitoring, BT Technology Journal, Vol 22, No 3, July 2004)

3.7.2. Place of fusion

- close to sensor less device, smarter sensors, hidden connectivity, higher energy usage
- close to center (context broker) less complexity, smaller costs, connectivity may be a problem
- hybrid schemes

3.7.3. Fusion and the consequences of the sensor failure

24h working regime

- continuous fusion and context building
  - continuous recognition of the ADLs
- evidence based, evidence weight changing in time
  - sensor design, Fault Tolerant techniques
  - sensor redundancy, alternatives
  - human as emergency sensor:
  - "discovering", "addressing", "interpreting", ...

3.7.4. Sensor networks

- local ad hoc - ...
- local, designed for a purpose

  - Smart House (Siemens, Phillips, ...)
  - general purpose (sensor networks, intelligent mote, ...)
- global ad hoc - Sensor Web

3.7.5. Technical challenges:

- ad hoc installation: generally no infrastructure on the installation area, e.g. in the woods - from the airplane task of a node: identifying connectivity and data distribution
- running on its own: no human interaction after installation, responsibility of the node to reconfigure after changes
- not supervised: no external power supply, finite energy supply. optimal energy spending: processing (less), communication (more), minimizing communication
- dynamic environment: adaptivity to variable connectivity (e.g. appearance of new nodes, disappearance of nodes), to variable environmental influences.

In the life of a sensor network the most important is

- energy spending battery, poor resources (dimensions, costs, ...)
- energy awareness:design and usage a single node (own task + router!) a group total network
- localization (establishing the space coordinates of the sensor)
  - GPS in external space, expensive (small cheap sensors), occlusions (dense foliage, ...)
  - recursive trilateration/ multilateration methods some nodes (higher levels of the hierarchy) with known location periodic direction senders, other nodes compute their positions

3.8. 2.8 AmI and AAC - Ambient Assisted Cognition, Cognitive Reinforcement

- supporting daily routine: taking medicine, cooking, handling household devices, hygiene,
- helping inter-human communication, keeping social contacts
- way finding, orientation for disabled people
- memory prosthesis
  - time-point based
    - fixed
    - relevant (relevant in a time-point, but can be delayed within acceptable limits)
  - event based
    - urgent
    - relevant
- increased feeling of safety

3.9. 2.9 Smart home and disabled inhabitants

- safety monitoring
  - systems and devices to prevent in-house accidents
  - monitoring - usage of water, gas, electrical energy ...e.g. automated switching on lights in case of getting out of bed at night
  - monitoring sounds at night - quality of night rest.
- social alarming (telecare)mixture of communication and sensory technologies - manual or automated signaling about the local need for the intervention of the remote care center
- healthcare, medical monitoring (telemedicine)telemedicine: direct clinical, preventive, diagnostic, and therapeutic services and treatments, consulting and follow-up services, remote monitoring of patients, rehabilitation services; patient education.
- comfortremote control for the disabled - lights, curtains, doors, ...more independence
- energy managementenergy saving services - timers, remote control ...
- multimedia and recreation
- communicationaudiovisual and communication devices for people with decreased mobility.

Basic critical functions/ services

monitoring fluid intake

monitoring meals

tracking location

detecting falls

3.9.1. Ambient interfaces for elderly people

web fastest layer of web user now 1:10 older than 60

by 2050, 1:5

by 2150, 1:3

majority of the elderly people are women (55%).

majority of the oldest people are women (65%).

- ageing effects: limited abilities, but with large diversity, no pattern.
- outage of the sensory abilities
  - senses (eyesight, hearing, sense of feeling, smell, taste, ...).
  - information processing abilities
  - intelligible speech
  - ability for precise movement
  - span of recollecting memories
- vision impairment sense of contrast, ...of the depth
- hearing disabilities

- feeling
- cognitive abilities
- Multi modal interfaces
  - general user picture, sound, touch
  - elderly picture, sound, touch
  - deaf picture, - - - , touch
  - with eyesight disability - - - , sound, touch

4. 3 Basics of embedded systems. System theory of embedded systems, characteristic components.

During the past decade the application of computing hardware went through significant changes: microprocessor-based systems became cheap and widely available and they are used in a very wide variety of devices nowadays. They not just got cheaper but their performance is increased significantly. Today's mobile phones and tablets are powerful enough to perform tasks that were only executable on personal computers a couple of years before. Consequently, these devices took notable market share from PCs.

Computing power in new environments and devices provided new perspectives for system developers: traditionally non-computerized devices gain new features and functions thanks to the embedded microprocessors. Our everyday devices (refrigerator, hifi set, television, vacuum cleaner, car, etc.) became more helpful and they provide new services. We can browse the web on our TV set, our car provides actual traffic information and by-pass roads, vacuum cleaners wander on the floor autonomously to keep it clean, our camera performs digital image processing in order to calculate the best settings for taking a photo, and washing machines adjust their program according to the weight and dirtiness of the cloths. Small computers are embedded everywhere in our environment.

All these new applications of microprocessors share the same characteristic: computing power is embedded in a hardware device in order to perform certain tasks in an application. Accordingly we can give the following definition for embedded system: a system that contains a computing element and operates autonomously in its environment in order to fulfill a certain purpose. They are not general-purpose computers but microprocessor-based systems designed to perform a certain set of tasks usually in a well defined, sometimes co-designed hardware environment. The software and the hardware are built together for a specific purpose and they form an embedded system.

Note : some authors prefer to define these systems as embedded computer systems to make a clear distinction between embedded systems in the 80s and 90s when microprocessors were not as wide used to build such systems.

4.1. 3.1 A basic anatomy of an embedded system

An embedded system contains two main parts: hardware (which is operating or embedded in an environment) and software (which runs on the hardware, or embedded in the hardware environment). These two parts must work in concert in order to create a successful embedded system, to fully utilize the possibilities of both worlds.

It is very hard to describe or design these systems in general: the wide variety of computing hardware (CPU, FPGA, DSP, special-purpose processing units, etc.), and the even greater selection of software architectures (networked computers, distributed systems, real-time systems, embedded operating systems and various kinds of technologies and pre-made components) yield very different designs.

The research of hardware-software co-design in the 90s aimed to create a unified model but only for a very limited set of possible hardware and software components/architectures. FPGA and DSP systems are good examples of this effort. Since then, the field of embedded systems grew greatly, nowadays there is a far greater set of possible hardware and software architectures.

Although DSP- or FPGA-based systems still fill in an important role in embedded systems, with the dawn of cheap, general-purpose microprocessors today's embedded systems are often resemble to conventional computer systems (from architectural perspective).

Instead of undertaking the impossible mission of creating an universal architecture for embedded systems researchers are helping system developers by defining the blueprints for such models. They define requirements, constraints and design concepts that system designers should take into account. For example, the ARTEMIS project funded by the European Union developed the GENESYS platform for this purpose. This platform characterizes the following architectural principles: complexity management (component-based, global time service, separate communication and computation), networking (basic message transfer and message types), dependability (fault containment, error containment, fault tolerance and security) and finally coexisting design methodologies (name spaces, model-based design, modular certification and coping with legacy systems). It also specifies a multilevel conceptual structure from the chip level to the system level, and categorizes architectural services into core, optional and domain-specific sets. Core services include identification, configuration, execution life-cycle, time and communication.

We will discuss the anatomy of an embedded system in more detail at two levels in this chapter: the application software designed for a certain purpose, and the embedded operating system which ease the development of the application software by providing a general software framework.

4.2. 3.2 Embedded software

Creating software for embedded systems is a key part in their development. Writing a program to run on an embedded processing unit might look similar today to developing applications in a conventional PC environment but in fact, it is more challenging. Embedded software must obey additional requirements like meeting deadlines during output generation, to fit into the available resource limits like memory and power consumption.

In the past these challenges were only solvable using low level programming languages. But developments in CPU architectures and language compilers, faster processing units and the spread of real-time operating systems have made the application of high-level programming languages (and related techniques) common.

Typical software building blocks in embedded systems include finite-state machines to build an event-driven reactive system, data stream processing and queues in order to cope with external input and waiting control structures.

A great deal of effort is put into program optimization in order to meet the specified resource limits. These include for example expression simplification, dead code elimination, procedure in-lining, register and memory allocation, etc. Fortunately, some of these methods are performed by program compilers. After developing the software the testing phase includes further performance and power analysis and optimization steps. Even the final size of the program must be checked to meet the allowed limits.

4.3. 3.3 Embedded operating systems

As embedded processors became powerful enough to handle more complex software the need to simplify the task of software developers led to the spread of embedded operating systems. Writing large applications performing multiple operations can be challenging without the support of an underlying operating system. The classical architecture of the personal computers moved into the embedded world: in order to handle multiple tasks at the same time, to ease the development and to increase the portability of embedded software special-purpose operating systems were designed.

Embedded operating systems perform similar tasks to their desktop relatives, but they also meet the requirements described in the previous section. They try to ensure that tasks are running within their allowed resource limits and they meet their deadlines. Real-time operating systems (RTOS) are specially designed to satisfy real-time requirements. They are widely used in complex embedded applications.

Although embedded operating systems resemble to desktop systems they very much differ in their internal architecture and operation. They have to meet additional requirements like predictability (with a calculated worst-case administrative overhead), they support time-triggered and event-triggered tasks, they support writing portable software with a (relatively) simple programming interface, they provide precise absolute and relative time for applications and they should have error detection mechanisms like tasks and interrupt monitoring.

The main RTOS components include task management (time-, event-triggered and mixed environments), communication (data exchange and coordination structures), input/output operations (time-constrained, non-blocking I/O, etc.), time service (clocks, alarms) and fault detection and handling (watchdogs, time limits, voting schemes, etc.).

Task management in RTOS typically handles time-triggered and event-triggered jobs. In the case of time-triggered jobs the temporal control structure is known a priory and the system can be designed accordingly. When the execution of tasks is evolving according to a dynamic application scenario event-triggered tasks can be used by the application developers. In this case the execution is determined by the dynamically evolving application scenario thus the scheduling decisions must be made during run-time.

In order to ease the life of application developers in a very heterogeneous world of embedded processors (and operating systems) a lot of effort has been made to standardize real-time operating systems. The IEEE POSIX 1003.13 standard for RTOS specifies several key areas like synchronous and asynchronous I/O, semaphores, shared memory, execution scheduling with real-time tasks, timers, inter-process communication, real-time files, etc.

Typical real-world RTOSes include Windows CE, VxWorks, RT-Linux, QNX and eCOS. See the "List of real time operating systems" Wikipedia page for a more current list. Some of the conventional, desktop operating systems also include real-time capabilities. Solaris provides real-time threads, Linux has a RTAI extension and 3rd party software companies provide RT extensions to desktop Windows systems.

5. 4 Basics of multiagent systems. Cooperativeness.

The heavily interconnected nature of today's computerized systems open a new era in computer software modeling and development. Today, even the smallest computer or sensor can be (and usually is) a part of a distributed system. Standardization processes helped greatly the interoperability and cooperation of these systems. Networked computers and peripherals, the client-server computing model and distributed systems facilitate the development of new kinds of techniques in software programming.

The increasing complexity of these networked, distributed systems led to the recognition, that conventional programming and modeling techniques are not always adequate. The continuously changing heterogeneous nature of the environment, the dynamic events in the distributed system require a new kind of "intelligent" behavior from the software systems.

Artificial intelligence, which goal is traditionally to develop intelligent systems, provided such a new computing paradigm called intelligent agent. In contrast to the prior standalone knowledge-based solutions these agents put an emphasis on the embedded nature of intelligent systems: they monitor the changes in their environment and autonomously react to them according to their goals.

5.1. 4.1 What is an agent?

An intelligent agent is an entity which has sensors and actuators, and based on its observations and goals it autonomously acts upon their environment. In practice they are software programs that react to the changes in their environment according to their programmed goals and behavior. In addition to these general primary features, they often exhibit more characteristics like being rational, truthful, knowledge-based, adaptive, etc.

There are several types of agents based on their characteristics and internals. Simple reflex agents simply percept the environment and act based on the current perceptions. Their operation is very similar to the traditional rule-based systems: if something happens then do something.

Model-based agents can deal with incomplete perceptions. They build a model of their environment, remember past events and states, and use these information to create their actions.

Goal-based, or deliberative agents add information about the desired world and internal states (goals) to the picture. They utilize search and planning algorithms in order to achieve these goals.

Real agents always have finite resources. They might have several conflicting goals and multiple ways for each goal to reach. In order to select the best achievable goal and the best way to reach it in a given situation and within the available resource limits, utility-based agents use a utility function. These rational agents try to maximize the utility function given the current state and the available resources.

Not all environments are known a priori and goals and situations reoccur time to time. Learning agents try to uncover knowledge from the past in order to perform better in recurring or unknown situations of the future. These agents have a so-called critic component that provides feedback to the learning component about the agents' operation in order to enhance the performance in future situations.

Intelligent agents seldom operate alone. In their environment there are other acting entities (e.g. other agents, software systems, humans, etc.). This implies that some of the agent's actions are related to communication with these active entities. When multiple agents are deliberately designed to communicate with each-other in order to solve a common task, a new kind of intelligent system is created: the so-called multi-agent system.

5.2. 4.2 Multi-agent systems

A multi-agent system (MAS) is composed of several intelligent agents which share a common environment and communicate with each-other in order to solve a common problem. These agents are typically software systems (might be running in some physical agent, like a robot), but it is not uncommon that humans also take part in such a MAS scenarios.

This community of agents can reach goals, perform actions, make perceptions that lie beyond the possibilities of individual agents. While they can achieve better results together they also arise new kind of problems to solve. In addition to the previously mentioned (local) agent characteristics and designs they require new kinds of features and methods to operate.

The first key component of a multi-agent system is communication. Agents in these systems should be socially active: they should communicate with others, ask or answer questions, make statements, etc. In order to achieve this common, standardized communication protocols and languages should be defined. Agents should share not only a common way to communicate but a shared knowledge about the communication in order to understand each-other.

The second key area of MAS development is the nature of cooperation. Communication is only a mean which can be used to share information about the environment and the state of agents and their problem solving. In order to solve a shared problem agents should cooperate with each-other.

5.3. 4.3 Agent communication

The definition of an agent states that it makes perceptions and performs actions on its environment. Thus, in agent theory, the communication also can be defined as a kind of action. According to the so-called speech act theory (which is part of linguistics research) entities communicate in order to make changes in others' thoughts, they perform communication like any other actions in their environment. Intelligent agents use this approach (in a simplified way) to model and implement communication. A set of so-called performatives determine what kinds of interactions an agent can have with others. These speech acts typically include queries, commands, assertions, refusals, etc. The content together with the performative forms the message an agent wants to convey.

As an action performed by agents, communication is based on two factors: a common understanding what the action means, and the method how the action is performed. To form a multi-agent system agents have to know how to form and interpret communication actions, and how to send those sentences to others. The former is defined by a so called content language, while the later is determined by communication protocols. These together form the agent communication language (ACL).

There is no generally accepted language for multi-agent systems. It depends on the actual application and the chosen tools of implementation. The most commonly accepted frameworks for such agent communication are the Knowledge Query Manipulation Language (KQML) developed by the DARPA Knowledge Sharing Effort and the Agent Communication Language (FIPA-ACL) by the Foundation for Intelligent Physical Agents (FIPA) consortium. Both systems share the same theoretical ground based on the speech act theory.

Agent communication protocols are high-level, layered systems based on the classical computer networking theories. The protocols are built upon on standard TCP/IP network stack at the transport level (usually TCP/IP, but it could be SMTP, IIOP, HTTP or other in order to ease the handling of firewall issues). Accordingly, they implement a message-oriented data transport mechanism. These protocols can be divided into three layer (above the transport layer): message, communication and content. The ACL message layer encodes a message that one agent wants to send to another. It will identify the transport protocol and specifies the performative that the sender attaches to the transported content. The communication layer defines common features for messages like the identity of the sender and recipient agents, and a unique identifier associated with the message (in order to refer to it in future messages). Finally, the content layer holds the actual content the agent wants to share with the recipient. It can be represented in various content languages like the Knowledge Interchange Format (KIF) typically used by KQML and also available in FIPA-ACL, Darpa Agent Markup Language (DAML) based on the Resource Descriptor Framework (RDF), FIPA Semantic Language (FIPA-SL), and many others.

5.4. 4.4 Agent cooperation

Autonomy and cooperation are key but conflicting characteristics of multi-agent systems. Reaching a common goal or performing a common action by individual entities means they give up a part of their independence (losing control) and engage in a cooperation with other agents (giving control to others). To ensure the success, agents should be deliberately designed to be able to take part in such scenarios.

In a multi-agent system cooperating agents take roles which serve as behavioral guidelines: they describe what are their expected activities and actions (including communication). A role of an agent can be known a priori or it can be determined run time using communication and some kind of advertising mechanism. Having the knowledge of agents' roles each individual agent can decide what it wants to be done by other agents (provide an information, perform an action, etc.). In order to facilitate this cooperation designers can deliberately introduce a set of messages to mediate these expectations: cooperation requests, direct orders, answers to these, and even sanctioning activities.

The nature of the cooperation varies very much depending on the application, the distribution and behavior of agents and the properties of the environment. Several methods were proposed at different levels of coordination including fully decentralized, event-driven systems, developing distributed algorithms for MAS implementation, behavior-based coordination, partially centralized systems to form virtual structures sharing global variables (like goals, measure of success, etc.), and so on. Even emotion-based control was proposed by some authors in order to simply cope with the complexity of agents' states and desires. Game theory put a lot of effort in researching this field and provides several approaches to choose from.

Designed to be cooperative is not always the best way to achieve certain goals or implement successful systems. In some situations, the details of cooperation cannot be specified exactly during the design of the system: environments change or they are simply too complex to cope with, uncertain events happen, the "winning strategy" is not known before or changing continuously, there could be "hostile" or competing systems, etc. In certain applications the success of a multi-agent systems might depend on many conflicting or even unknown factors thus making the prior design very hard or even impossible.

To cope with such applications and environments we can design multiple agents based on different assumptions and operation model for similar purposes, and we can build a multi-agent system where agents (at least partly) compete with each-other. In some cases competition is the heart of the problem. For example, a main application area of competing agents is Game AI (especially networked games). Other typical examples include trading agents, distributed resource allocation and supply-chain management.

Developing competitive agents in a MAS environment requires the specification of a success measure which is often done in the form of rewards and penalties. These are typically used in the learning component of the agent in order to increase its effectiveness: maximize rewards and minimize penalties. Competitive agents can also be cooperative in some situations or in a set of agents. In addition to local measures, system developers could also introduce global rewards (e.g. for a set of agents) in order to tailor agent behavior between competition and cooperation in a mixed environment. An example could be a robot soccer game where team members typically cooperate in order to win but agents will compete with the other team and sometimes even with their own associates in order to score a goal.

6. 5 Intelligence for cooperativeness. Autonomy, intelligent scheduling and resource management.

In the previous chapter we learned about communication and cooperation between intelligent entities (agents) in general. Communication was defined as a kind of action to query, command, ask, answer, etc. others. In order to reach a common goal or to perform a common action individual entities give up a part of their independence (losing control) and engage in a cooperation with other agents (giving control to others). To ensure the success, agents should be deliberately designed to be able to take part in such scenarios.

The nature of the cooperation varies very much depending on the application, the distribution and behavior of intelligent entities and the properties of the environment. In this chapter we examine this area through the problem of resource allocation and dynamic scheduling in embedded systems.

6.1. 5.1 Intelligent scheduling and resource allocation

In embedded systems limited resources and the real-time nature of the required actions and the environment are important characteristics. Traditional scheduling (control) systems were developed according to a top-down process of command and response and they relied on a centralized architecture. In this architecture a central entity was responsible for maintaining scheduling information and performing task and resource allocation accordingly.

These systems have several weaknesses. On one hand the existence of a centralized unit is a performance bottleneck, on the other hand it is a critical single point of failure. As the complexity of the environment and performed tasks grows, the hierarchical scheduling system becomes increasingly complex, expensive and time consuming to develop. Furthermore, the distributed nature of the environment and the embedded system introduces latency, communication bottlenecks and new possibilities for failure. It became clear that these systems may perform badly in more complex environments and tasks.

Today's scheduling systems should be more adaptive, flexible, efficient users of resources and robust against failures. These properties are in pair with the characteristics of intelligent agent systems thus the application of such systems are a promising approach to solve scheduling and resource allocation tasks.

6.2. 5.2 Cooperative distributed scheduling of tasks and the contract net protocol

One of the first and most widely used form of agent-based scheduling is a fully distributed architecture of cooperating intelligent agents. In these systems agents represent entities of the environment and the task at hand such as resources, jobs, etc. Thus they have the ability to act locally (independently of any other entities) and to cooperate directly with related entities globally. Local actions ensure the reactiveness, simplicity of local planning and they make for the global fault tolerance of the entire system. When cooperation is needed (e.g. an agent performing a job requires a resource to complete its task) agents interact with each-other directly, they plan their actions according to this cooperation.

he key problem of cooperative distributed scheduling systems is that agents make decisions based on their knowledge about the outer world, and they local view might be incomplete, out-of-date or even inconsistent with others' knowledge. Several protocols exists for sharing information and for cooperation between distributed intelligent agents. One of the earliest solution is the contract net protocol. It was developed in the early 90s originally as a high level protocol among the nodes of a distributed problem solver. The protocol can be used for allocating tasks in the network: exchanging information between interested parties, evaluating the current situation and making an agreement by mutual selection of the best way of action.

Individual nodes also plan their activities though contracts. An agent may decompose its contract into subtasks to be performed by other agents. After the decomposition is done, the agent announces its subtasks in the system and waits for other agents willing to perform them. The contract net protocol is used to gather information from other parties about their bids, to clarify the details of a subcontract and to establish mutual agreements with the subcontractors. The agent distributes its subtasks according to the subcontracts, monitors their execution and receives their results. Usually, there is a special master agent that receives tasks from outside of the system. There are several methods how the agent can select the best performing subcontractor for a given task. It can be even an iterative process during which the agent may decide to perform the decomposition in a different way and repeat the bidding part. This way it can select the best decomposition that fits the task and the actual set of available problem solvers (subcontractors). Thus the entire system approaches an optimal solution and it becomes more adaptive to changing situations (even to failures).

Some systems even allow the termination of a contract. Events during the execution of contracted tasks might yield to failure, or sub-optimal execution. Specifying a suitable termination penalty even makes possible to reconfigure the contract network on-the-fly to be more optimal when the situation changes (new events happen, better contractors become available, etc.). In such situations agents might break agreements (and pay the termination penalty), and make new contracts to get better results overall.

The contract net protocol is used in many application areas, for example distributed sensor networks, intelligent manufacturing systems, grid resource negotiation, service composition in cloud computing, etc. It is best suited for applications when a very limited number of projects (task compositions) are given to certain individual agents (in contrast to the entire network), there are well defined tasks decomposition methods, and the roles of agents can be planned accordingly.

6.3. 5.3 Market-based approaches to distributed scheduling

When there are many projects (tasks), and resources and processing units are distributed across a system of many individual intelligent entities sharing the necessary information in order to make contracts between interested parties becomes increasingly hard. Therefore some multi-agent cooperation systems specify a market of resources, projects, tasks and bids and they usually introduce auctions and currency mechanisms. In contrast to the contract net protocol this scheme suits multi-project applications better. In these scenarios there are several projects in the system, and these are assigned to many different agents.

Manager agents (responsible for projects) decompose their work into tasks and they create a market of job requests. Task agents acquire jobs from this market. They are also looking for resource agents to provide the necessary resources to finish their work.. If they need help from other agents they might take a procurer (coordinator, or manager) role, and they announce subtasks on the market to be taken by other task agents. Resource agents announce their resources on the market, monitor other agents' resources, market supply, demand and prices, and wait for actual orders from task agents. There might be special coordinator agents in the system whose role is to accept projects from outer systems, manage and monitor their execution.

In a market-based system every contract is made based on price negotiations. Resources and tasks have list prices determined by their governing (performing) agent but this price is negotiable in a one-to-one or many-to-many contracting scheme. Every agent is in charge of its projects, tasks or resources, and it determines their prices according to the market situation and to its internal rules and goals. There are several approaches how agents can determine the price of performing a task or providing a resource. The typical price negotiation method is some kind of an auction scheme.

Price-negotiation auctions may concern single or multiple units (e.g. resources or jobs), and their goal is to establish an agreement between contractors on the price of the units in the contract. Auctions have rules determining unit allocation and prices based on the bids of the bidding participants. They usually start from an initial price and increase or decrease it to a level when only a single winner exists for each given unit. Price equilibrium is a set of prices for the considered units and agents when each agent maximizes its utility according to the determined unit prices. There might be several equilibria in a given situation; it depends very much on the characteristics of units, auction mechanism and agents bidding policies. For example, in discrete allocation problems it can be shown that price equilibrium is an optimal solution to the problem. It can also be proven that for single-unit scheduling problems there exists globally optimal price equilibrium. The question is what kind of auction mechanism will lead to a price equilibrium (or optimal solution) in a given situation. There is a broad class of auction mechanisms. They can be classified according to many characteristics including the set of participating parties, price determination, matching algorithm, execution algorithm and timing, bid and price restrictions, openness, etc.

One of the most important parameter is that the auction considers a single unit, or a set of units. It is shown that this very simple single unit ascending auction can produce unit allocations that are not convergent to the optimal solution, therefore researchers proposed combinatorial auctions where agents submit offers for a set of units. Project manager agents usually require several tasks to be done and a set of resources to support them, therefore it is also more natural to them to acquire the needed units in a (few) combined auction(s). During the iterative combinatorial auction process the bidding agent announces its bid as a pair of unit-set and price. The auctioneer collects all bids and calculates a provisional allocation of units among the winning bidders maximizing its selling prices. Then it announces this allocation among all bidders and it gives a possibility for loosing agents to resubmit their bids (with higher prices). Supposed that there are enough units for all bidding agents this process will end when each bidder becomes a winner and all unit bundles are allocated.

The other interesting property is the bidding system: the rules and strategies to determine unit prices. Ascending auctions determine an initial price for a unit, then this price is increased by the bidders by predefined discrete amounts. A simple bidding strategy could be the following. Agents start bidding for a set of units at initial price levels. When they lose one or more bids (the unit price went higher than accepted), they may change the unit set and their bids. Their goal is to finish the auction with a positive surplus while winning as many bids as possible. One of the descending-price versions is the Holland auction. In this scheme the auctioneer starts with a price much higher than the expected market price and waits for bids. When no bids arrive it decreases this initial price. This is continued until a bid is made or the price drops below the tolerance level of the auctioneer. This implies that bidders can value units before the auction starts. There are auctions when the bidders do not know anything from the auctioneer and from other bidders.

The Vickrey auction is a second-bid closed auction, where every agent secretly sends its bid to the auctioneer, the highest price wins, but the winning price is set according to the second highest bid. This scheme is very simple, it can be used in a wide variety of applications, it has a low cost of bidding (zero payment for losing bidders), and it promises higher revenue for the auctioneer (it is proven that its average revenue is not less than in any other efficient mechanisms). These good characteristics make the Vickrey auction very popular.

Evaluating the market-based scheme we can list several advantages. It is naturally decentralized thus it suits well applications when there are many processing units, resources and tasks distributed over a large system. Communication is limited to unit and price information exchange. This simplicity helps to acquire good performance and high reliability for the entire system. Properly selected and tailored auction mechanisms may yield to Pareto optimal solutions in certain application scenarios where a profit of one agent cannot be increased without decreasing another agent's profit. These systems also have some drawbacks. The market-based approach is non-deterministic, it could even lead to uncertain situations where the outcome could also be extreme. This very much depends on the characteristics of the scheduling problem and the auction scheme and bidding strategies chosen. This kind of systems is, however, proven to be very effective in larger scale manufacturing and trading applications.

6.4. 5.4 Other intelligent techniques for scheduling

Several researchers proposed additional techniques for dynamic intelligent scheduling. These techniques include several artificial intelligence methods including - but not limited to - learning (e.g. using artificial neural networks), fuzzy logic, Petri nets, knowledge-based systems, etc., and techniques from other fields such as, for example, game theory.

Knowledge-based approaches try to include the technical expertise or experience of the specified domain in the scheduling algorithm. This knowledge can be represented as a constrained search problem, where the algorithm tries to find an optimal solution (path) given the enforced constrains.

Some early systems used the blackboard architecture somewhat similarly to the agent-based approach. Problem solving entities shared their information and progress using a centralized information storage (blackboard).

Simulation-based scheduling tries to estimate the progress of the negotiation process in different applications. At a stage of the negotiation (contracts or auctions) agents could better evaluate the current situation by simulating other agents actions for the next stage(s). Based on these information they can revise their strategies for the following stage. This process can also be repeated for a couple of stages to determine the best final strategy.

Hybrid systems were also developed by researchers, where knowledge-based scheduling used information from a simulation module to evaluate different possibilities, and learning or genetic algorithms tailored the parameters of these modules.

Handling uncertainty was possible in some systems using fuzzy logic techniques. Uncertain lengths of resource shortages and operation disruption periods were the main reason for introducing fuzzy logic-based decision support to determine when to reschedule the system instead of simply waiting for their end.

The other source of problems in multi-agent scheduling is the handling of conflicts. Sometimes it is even non-trivial to recognize conflicts, then it should be examined and classified before a conflict resolution strategy can take place. The nature of conflict resolution depends very much on the application domain and the chosen agent cooperation strategies (cooperative or competitive).

Game theory models are usually too simple to follow by competitive conflict resolution but they provide a very good ground to illustrate the problems and possible outcomes of certain situations. They also enlighten the conflict of creating a globally optimal system from locally optimal agents.

7. 6 Agent-human interactions. Agents for ambient intelligence.

7.1. 6.1 Multi Agent Systems (MAS) - advanced topics

7.1.1. Logical models and emotions.

MAS environment creates additional challenges for handling knowledge. For an effective communication and load sharing MAS agents must reason not only about the task environment, but also about themselves (their task solving capabilities) and about the cooperative/competitive profiles of the other agents. In addition, reasoning of an individual agent is isolated and perhaps not entirely globally correct. Limited resources, limited sensory data, uncertainties, can result in situations where facts believed by an agent will diverge from the objective view of the world. It is thus important to distinguish between agent beliefs and the true facts. The problem is generic and is amplified in the MAS setting, where the beliefs of the cooperating agents should be somehow mutually consistent.

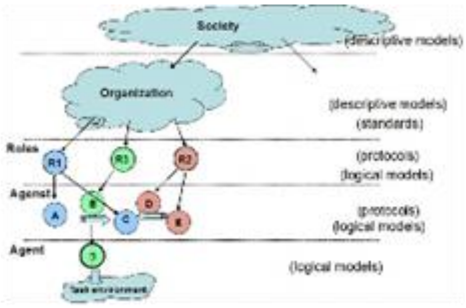
To solve the problem technically various modal logics are used in the MAS theories to model agents' beliefs and goals, with modal operators equipped with so called possible world semantics. The ruling agent model within this framework is so called BDI (Belief-Desire-Intention) model structuring agent knowledge into beliefs (factual knowledge), desires (long range goals), and intentions (action plans directed toward goals and conditioned on the beliefs). The BDI models were recently extended with the description of emotions (EBDI). Emotional information represents a highly concise system state information and makes it possible to describe very simply very complicated systems and to predict their future behaviour on this basis.

7.1.2. Organizations and cooperation protocols.

MAS systems can acquire diverse organizational forms. The literature on the MAS investigates organizations of hierarchy, holarchy, coalition, congregation, community, federation, team, market, matrix, and various hybrid forms. Organizations and logical models help to clarify the source and the way to handle the conflicts. If the organization is tightly knitted and designed as a whole (e.g. a team), there is less prospect for conflicts, albeit even here conflicts are possible.

Co-existence within an organization calls for a highly organized way of information exchange between agents and this can be shaped only based on the human conversation. The very first agent communication language (ACL) was KQML. It introduced communication model based on human speech acts, which became de facto standard in the agent communication design. Recently developed agent organization and communication semi standard FIPA defines the communication language (FIPA ACL) as a multi level structure, where the message type level language is composed from the standardized speech acts, and the message content level language is a relatively free choice tailored to the freedom of the designers. To help in the agent design message types are provided with semantic descriptors expressed in modal logic.

Well interpretable communication language serves as a basis to design knowledge intensive communication protocols. Such protocols provide the framework for the task sharing and load coordination (conflict resolution included), cooperative reasoning (cooperative learning included), and cooperative planning. To the widely used protocol assortment belong the protocols: Master/Slave, Contract Nets, protocols for brokerage, arbitrage, cooperative (forward and backward chaining) reasoning, various auction, and voting protocols



16. ábra. Hierarchy of MAS models

7.2. 6.2 Embedded Intelligent System - Visible and Invisible Agents

Ambient intelligent environment - intelligent space - implies a long term co-existence of software and the human agents and calls for a number of various agents, performing services to the intelligent space on the whole, in particular to the human population and the equipment essential for the environmental status quo.

Along this life line every agent has its role and is immersed in its natural environment. Software agents are immersed in the virtual world, human agents in the physical space enhanced by intelligent artifacts and the presence of ambient intelligence. Every group of agents requires different kind of organization with formally or informally defined roles and structure. The organization of software agents is well defined, purposefully designed and usually represents a team of specialists (e.g. device-agents governing domotics) with elements of hierarchy defined along the prescribed responsibilities. Organizations in the human agent world are less set, the roles are not always well defined (e.g. a visitor), but also here we have definite specialists and hierarchy (e.g. a cleaning worker, a health care worker, a nurse, a doctor).

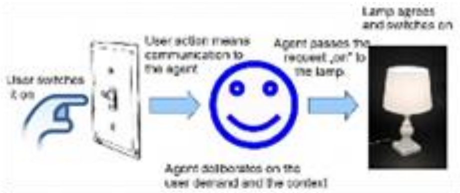
To persist in organizations agents must communicate and cooperate with other agents, including human agents, acquire information from sensors and influence the environment (ambient properties and the state of the agents) by various effectors, including communication commands, requests, or signs, see Fig.18.

Agents in their actions can be:

- device-oriented: i.e. they are bound to the devices for a given duration of time and they handle them or represent, facilitating their usage to the human users, or adding value to their functions. Devices can represent signal flow into the physical environment (heaters, curtains, TV, water tap, medicine dispenser, etc.) or signal flow from the physical environment (sensors, device states, context data, etc.).
- network-oriented: i.e. focused on tasks constituting a part of more involved cooperative interactions towards some global goals, communicating with other agents.

7.2.1. Artifacts in the Intelligent Environment

The general technological advancement which introduced considerable computing capacity into the majority of the domotics (widely understood household equipment) and integrated them with just as easily introduced networking means redefined the causal chain from the user's goals to the functional setting of the responsible equipment, see Fig.17.



17. ábra. „Agent-in-between" - decoupling the human user from the hardware environment.

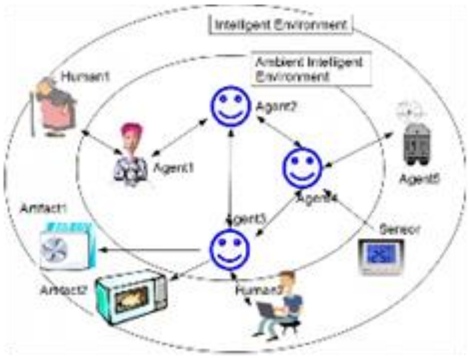
Originally the integrity of the chain (like e.g. form the wall switch to the lighting bulb) secured the proper functioning, and the full responsibility for the course of action and the results rested on the user' sound judgment and decision. Automation made it possible to introduce mediating intelligent systems which could in principle counteract the user decision. Is it a positive, essential development, yes, or no? And why?

Of course yes! Decoupling the human decision from the execution of setting a particular function in a particular equipment opens opportunities for:

- placing the user's decision within the wider context (known to the agent from context awareness) and then to reconsider its execution,
- redirecting the user's demand to other, functionally equivalent equipment,
- overriding the user's decision if considered inappropriate or harmful (e.g. a small child trying to switch on the TV, or to open a fridge),
- storing the context of the user's demand for further processing (learning user behaviour).

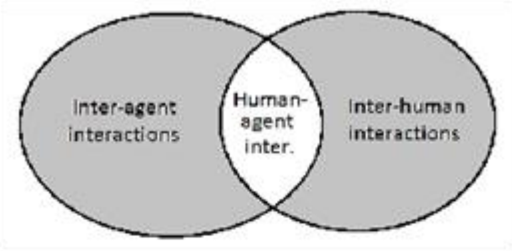
7.3. 6.3 Human - Agent Interactions

Both (software and human) agent societies interact, because they work towards a common goal (maintaining a "healthy" state of the embedded environment). Inter-agent interactions happen accordingly to strictly designed and executed protocols, using specific message languages (Agent Communication Languages). Inter-human interactions follow the well known schemes of informal natural language conversations, news, announcements, or written messages.



18. ábra. Population of intelligent agents in an intelligent embedded system.

The most challenging in the realm of the human-agent interactions, as the medium where organizations, roles, and individuals, not necessarily "designed" to cooperate, meet and interact together. The complexity of problems calls for sophisticated agent organization, with a full spectrum of possible interactions, backed up by suitable interface designs.



19. ábra. Realm of agent interactions.

In the following we will shortly analyze the problem of human-agent interactions from various, generally independent aspects, called "dimensions", but together yielding the complex picture of intelligent space interactions.

7.3.1. 6.3.1 The group dimension - groups of collaborating users and agents.

Agents can interact:

- one-to-one: e.g. a human agent asking an intelligent fridge about the expiration date of the milk, asking the personal secretary agent about the date of a meeting, etc.
- one-to-many: e.g. an agent sending out call for proposal, recruiting the help of other agents, to chose finally the single agent being asked to actually execute the request (so called Contract Nets protocol).
- many-to-one: e.g. many information providers converging on a particular user, human agent preparing to start for the work and obtaining the information about the weather, traffic, work schedule, things to do after work, etc.
- many-to-many: a typical interaction scheme within the ambient intelligence space, where device-oriented agents communicate with network-oriented agents to jointly maintain context information and on this basis the context sensitive maintenance of the intelligent space.

There are several possibilities for communicating within the group:

- multicast - messages are sent to every member at once (every member of the group must be known locally and there must be a route between all members).
- group-neighbors - an agent sends a message to its group neighbors, that they in turn send it to their group neighbors and so on. It is particularly useful in organizations (networks) established in an ad hoc way because it provides a way of using local knowledge about the network.
- broadcast - messages are sent to every neighbor, not just group neighbors. The neighbors may or may not forward the message.

7.3.2. 6.3.2 Explicit - implicit dimension.

The interaction between agents (humans included) can happen via two general "meta modalities":

explicit interaction can be:

- classical interaction means using traditional computer peripherals, like keyboard, mouse, screen, etc. It suits only technologically apt and healthy persons, and as a rule, is not the primary mode of interaction in the embedded intelligent systems.
- in artifact-based interaction an agent calling for human attention will use the interface of the equipment or some artifact of mixed functionality, like e.g. blinking lamp on the microwave, calling with a message via the mobile, or turning the digitally controlled picture on the wall into an optional text screen.

Contrary, the idea of implicit interaction means no direct purposeful contact (using some equipment) between the agent and the user. The agent is affecting the environment and expects the human user to be aware of it after a while (via his or her natural sensors), to interpret it correctly and to do something, which the agent wanted to achieve, but without an explicit communication action.

7.3.3. 6.3.3 Time dimension - Time to interact and time to act.

The conditions relating the temporary dimension of the interactions can be summarized in the following table:

	Short action	Durable action
Short interaction	(1)	(2)
Long interaction	(3)	(4)

3. táblázat.

(1) short interaction with the following immediately executed action means usually an unambiguous equipment related command or request for information already available in the context database.

(2) short interaction can command an initialization of a process or even round the clock activity. In this case the interacted command must be in the sequel continuously anew interpreted and the execution of the action will happen along changing contexts, putting additional computational an inferential burden on the software agents.

(3)-(4) long interactions means usually ambiguous information exchange calling for a lengthy dialogue to reach mutual understanding of real issues and relevant information. As the time span of the interactions can extend well over the horizon of the changing context, its interpretation with respect to the purpose of the interaction will burden both the software and the human agent, being possibly equally difficult to both.

7.3.4. 6.3.4 Roles vs. roles.

The very idea of an agent is related to the delegation of the burden (task). The human agent usually asks the software agent to do something instead of himself, hoping to have it faster, better, or just wanting to free his faculties from some mundane task. The great variety of knowledge intensive skills implementable in the artificial agents makes the delegation similarly flexible and adjustable, leading to a variety of possible agent roles.

20. ábra. The role of the artificial - the role of the natural.

The lower levels of the possible rendering of the roles of software and human agents belong already to the every-day practice, but some explanation is due regarding the upper levels of Fig.20. A human inhabitant of the ambient intelligent space (not the "computer user", but the "space user") is at the same time an essential part of the context information (who, where, how behaving, what state of health and mind, etc.), but also a highly perceptive and intelligent mobile sensor platform, equipped with sensors and processing very difficult to reproduce artificially ("something burning in the kitchen", "it looks like rain", etc.). Even more than that, it is also a "mobile robot" with exceptional, however limited action skills ("please help John to take the book from the upper shelf", "please go to the bathroom and turn off the tap", etc.). The only (and very difficult) issue is to ask for the information or action in a natural way acceptable to the human agent, which means almost exclusively some form of written or spoken language.

At the highest level the "suspect" means that the human agent is an observable to the software agent which builds human-related context information to act on it later in human interests. A human can be easily measured if the required information is low level, but it can be formidable processing problem if the sought information is related to the human intents and goals, and the human agent is not co-operating in the interaction.

7.3.5. 6.3.5 Modalities of Human-Agent Interactions

The capability to know how to interact with the user for a given demand and within a given context assumes that the agent has (or can choose from) suitable modalities and interfacing means for interactions. We will review now what assortment of tools can we list at the agent disposal. In this we will focus on new, nontraditional ways of expressing interactions, stepping over the accustomed ways and means of human-computer interactions of technically apt, professional users.

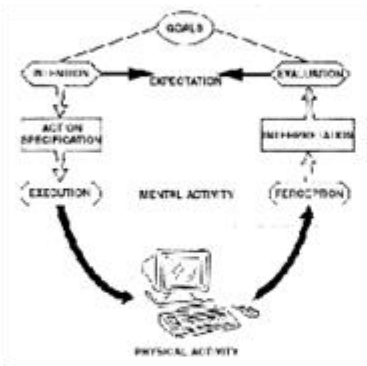
- Unimodal HAI systems (unimodal interaction addresses only a single sensory channel of the human user)
  - Visual-based systems,
  - Audio-based systems,
  - Sensor-based systems.
- Multimodal HAI systems (by definition it addresses more that a single sensory channel of the human user, passing the information in parallel, duplicating, enhancing, or complementing it)
- Visual-based HAI
  - Analysis of facial expressions, (for emotional identification or decoding anticipation),
  - Tracking body movement (distant, e.g. Kinect-tracking, identifying postures),
  - Gesture recognition, (for direct control),
  - Gaze detection (tracking the eyes movement on the screen in case of users handicapped in other means of interaction, see also brain-computer interfaces).
- Audio-based HAI
  - Speech recognition (with the focus on topical understanding),
  - Speaker recognition (with the focus on individual identification),
  - Auditory emotion analysis (the old Stress Detector),
  - Human-made noise/sign detections (gasping, sighing, laughing, crying, snoring, but also sound originated in some human actions, like running water from the opened tap, squeaking of the opened door, sound of falling, etc.),
  - Musical interaction.
- Sensor-Based HCI
  - Pen-based interaction
  - Mouse and keyboard
  - Joysticks

- Motion tracking sensors and digitizers
- Haptic sensors
- Pressure sensors
- Taste/smell sensors
- Brain-computer interactions (important, if technologically mature, for heavily handicapped users to pinpoint user's intents and emotions, with the most important: (1) decoding human awareness of computer error, (2) decoding human anticipation to upcoming events.

7.3.6. 6.3.6 Structure of Human-Agent Interactions

Whether human-to-agent, or agent-to-human interaction is considered there is a definite sequence of mental/physical activities structuring the interactions (see D. A. Norman and S. W. Draper, User Centered System Design: New Perspectives on Human-Computer Interaction, NJ: Lawrence Erlbaum Associates, 1986.)

Seven stages of action. An individual considering interaction (1) must have some concrete goal in mind, from which it infers that interaction is to be expected. (2) Operationally goals give raise to intentions, which in mental agent models (e.g. the BDI model) are precursor to actions. (3) Then actions are conceived (meaning the choice of the interface, the protocol, the message content, etc.) and (4) executed. The execution should be coupled with some follow up, (5) ensuring the possibility for a feed-back perception (like e.g. a click sound when typing). (6) Percepts must be then interpreted within the perspective of intents and goals and on this basis (7) the success fulness of the interaction is evaluated.



21. ábra. Seven stages of action in human-agent interactions, after (Norman, 1986)

The distance (the gap) between the goal and the physical activity leading in principle to achieve this goal is termed the gulf of execution (left side). In other words, the gulf of execution is the difference between the intentions of the users and what the system allows them to do or how well the system supports those actions.

The gulf of evaluation (right side) shows the degree to which the system or artifact provides representation that can be directly perceived and interpreted by the user (with respect to his expectations and intentions). It means that the gulf of evaluation measures the difficulty of estimating the state of the system and how well the artifact involved in interaction supports the discovery and interpretation of that state. Summing up, the gulfs of evaluation and of execution are the discrepancy between human's internal goals and his expectations and the availability of information specifying the state of the environment and how it may be changed.

7.3.7. 6.3.7 Dialogues

A dialogue is an exchange of speech acts (asserting, questioning, refusing, etc.) between two interacting partners in turn-taking sequence aimed at a collective goal (N.b. speech acts need not to mean real speech but any form of communication based on messages designed acc. to the speech act theory). The dialogue is coherent to the extent that the individual speech acts fit together to contribute to this goal. Each participant has an individual goal in the dialogue, and both participants have an obligation in the dialogue, defined by the nature of their collective and individual goals.

The following is Walton and Krabbe classification of types and subtypes of dialogues.

Dialogue type	Initial situation	Goals of participants	Goals of dialogue
Persuasion	Conflict of opinions	Persuade the party	Resolve/ clarify issu

		proved, at the present state of knowledge.	retraction once set in place at the appropriat

Traditional Human-Computer Interactions lack the eristic dialogue, where it can appear at the most on the side of under-educated and exasperated user calling names the unforgiving hardware, because it cannot read his intentions and correct his evident mismanagement of the interfaces. Situation is quite different in e.g. AAL applications, where during Human-Agent Interactions we do expect the user to be technologically nadve, inapt, clumsy, and not motivated and easily turning hostile, when faced with too large gulfs of execution or evaluation. Here "reading intentions" and "correcting user interactions" is a very hot research topic with plenty to gain if the results mature.

7.4. 6.4 Interface agents - our roommates from the virtual world

Partners to the human agents and the responsible actors in the human-agent interactions are the interface agents. By definition aware of the human context (presence, location, preferred modalities, expected activities, and the like) and designed to make most of it in the interest of successful interaction.

7.4.1. Issues and challenges

The basic requirements for a successful interface agent design are:

- the agent must know who (where, when, why, ...) is the user,
- the agent must be able to interact with the user (via the interface and modality convenient in a given context),
- the agent must be competent (knowledgeable) in helping the user.

These requirements revive plenty of serious challenges when trying to learn about users:

- identifying users' goals and intentions based on observations and user's feedback,
- acquiring sufficiently rich context to interpret the users' goals,
- self-adapting to the user's changing objectives,
- do it all efficiently.

Designing interactions with the user presents the following challenges:

- to decide how much control can be delegated to the agent,
- to build trust in the agent's actions or advices,
- to place the technical interaction within a metaphor understandable and acceptable to the user,
- making the interaction simple to comprehend and to participate in.

Competence is the hardest problem of all, because it is heavily knowledge intensive. After the agent inferred what the user is doing (how to do it?) and maintains an appropriate form of interaction, it must still design a plan of action (physical artifact-based actions, explicit interactions, implicit interactions) that will truly help the user. Here we must equip the agent with the capability of:

- knowing when and how to interrupt the user, if needed,
- performing tasks autonomously but in the way satisfying the user,
- at least partial automation of tasks (for more frequent user's demands).

7.4.2. Interface agent systems - a rough classification

A very short and clearly non-exclusive taxonomy of such agents, "visible" to the user can be as follows:

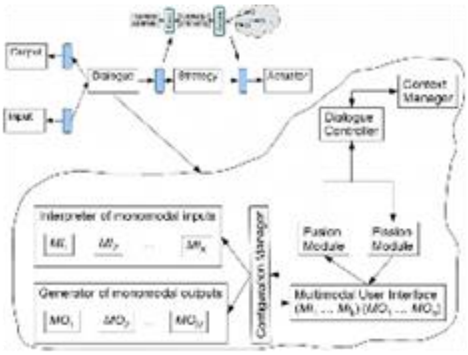
- character-based agents (e.g. conversational agents with personality),
- social agents (helping the user in social tasks), systems,
  - e-commerce systems,
  - entertainer systems,
  - expert assistance systems,
  - meeting schedulers, etc.
- agents that learn about the user
  - monitoring user behaviour (see implicit interactions)
  - receiving user feedback
  - explicit feedback
  - initial training set
  - programmed by user (usually in basic artifact-based interactions)
- agents with user models (all these models can be a part of retrievable context information, but require different handling at the agent's level)
  - behavioural model
  - knowledge-based model
  - stereotypes.

7.5. 6.5 Multimodal interactions in the AAL

7.5.1. Multimodal HCI Systems

Multimodal interaction appears already even in the traditional computer set-ups, where the visual information appearing on the screen is accompanied by the keyboard click or the sound of a beeper. But basically only these two are used, in fairly simple situations.

In the intelligent embedded environment, as a rule, the presence of a variety of heterogeneous, sometimes even ad hoc interfacing devices (see the spectrum of possible interactions) is coupled to the fact that the user society is populated not by professionals but by naive, lay, under trained, interaction-handicapped, or even technology-hostile individuals. In consequence the usage of the proper interfaces and devices is context dependent and thus should adaptively vary according to a permanently changing context within the users environment (e.g. the user moves through his house while he is exposed to a permanent change of context regarding the usable equipment and tools of interactions). So the choice of the interface, the choice of the content and the form of the message must be subject of a thorough and information intensive planning, see Fig.20.



22. ábra. Interface agent with multimodal interface management capability (MI - multimodal input, MO - multimodal output, Fusion/ Fission - composing/decomposing multimodal message from/into its monomodal components).

Systems supporting multimodal interactions (equipped possibly with a modality to pass over and to generate emotional information) are the basic requirement in smart video conferencing, intelligent homes/offices, driver monitoring, intelligent games, helping people with disabilities with advanced robotics, advanced healthcare systems, etc.

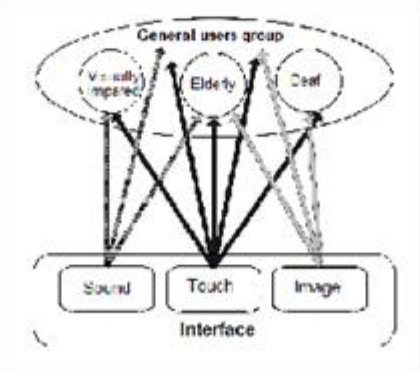
7.5.2. Interfaces for Elderly People at Home

The aging process is generally associated with a decrease in functionality and sensory loss, as well as cognitive slowdown. One of the most obvious effects of ageing is sensory loss. For many older people, some abilities tend to remain intact or slightly affected, while others suffer from a great deal of loss. This depends greatly on the individual.

The decrease in functional ability can affect the following adversely:

- Sensory organs (vision, hearing, smell, tactile sensation, taste, etc.)
  - External sensory loss:
- Vision loss
  - Hearing loss
  - Tactile sensitivity loss
  - Taste loss
  - Internal sensory loss
- Loss of thirst sensation
- Actuators
  - Loss of dexterity
  - Loss of safe and effective agility, reduced speed and increased variance in the timing of precise movements.
  - Speech intelligibility
- Cognitive Decline
  - The information process capacity.
  - Length of time required to retrieve information from memory.

7.5.3. Multimodality and Ambient Interfaces as a Solution



23. ábra. Multimodal interface for an interface agent interacting with handicapped users.

Simultaneous or configurable multimodality can improve user interaction and reduce greatly both gulfs of interaction. The design can take advantage of the different senses to transmit information in a more unambiguous, unobscured way.

8. 7 Intelligent sensor networks. System theoretical review.

Sensor network technology is the result of combination of miniaturized autonomous sensors with communication technology. In some cases wired network communication is applied, but typical sensor networks are built using wireless communication technologies. The most common wireless solution uses radio communication but optical sensor networks and in underwater networks with acoustic communication are used as well.

Wireless sensor networks (WSNs) have some or all of the following properties:

1. wireless, ad hoc communication,
2. mobility, topology changes,
3. energy limitations,
4. spatial distribution of sensors and computational resources.

Sensor networks are used in several fields. According to a recent survey the following applications are the most frequent fields (in decreasing order of answer counts in the survey):

- Basic scientific research
- Medicine, healthcare
- Automotive
- Manufacturing
- Chemical industry (petroleum, gas etc.)
- Test facility, metrology, compliance
- Power generation and distribution
- Aerospace

- Food, pharmaceutical
- Buildings, structures, HVAC
- Consumer products
- Transportation - trucking, railroad
- Agricultural machinery, vehicles
- Security - homeland, local (e.g. police and fire)
- Civil infrastructure, municipal services
- Military (not incl. military aerospace)
- Ecology, biology
- Transportation - ships, boats, ferries etc.
- Etc.

The ranking is probably not a very reliable one, but the importance of the sensor networks is apparent in these several diverse fields.

A wireless sensor network (WSN) consists of spatially distributed autonomous sensors to monitor physical or environmental conditions, such as temperature, sound, pressure, etc. and to cooperatively pass their data through the network to a main location. The more modern networks are bi-directional, also enabling control of sensor activity. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance; today such networks are used in many industrial and consumer applications, such as industrial process monitoring and control, machine health monitoring, and so on.

The WSN is built of "nodes" - from a few to several hundreds or even thousands, where each node is connected to one (or sometimes several) sensors.. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth.

8.1. 7.1 Basic challenges in sensor networks

There are several challenges, which have to be solved. Some problems are so problematic that intelligent methods are needed to solve them. The most important challenges are the following:

Energy consumption. Usually energy consumption needed is provided by batteries or using some form of energy harvesting.

Communication. Sensor networks have low-rate data and many-to-one data-flow. The end-to-end routing schemes for conventional networks like mobile ad-hoc networks are not appropriate for sensor networks. Data-centric technologies are needed, getting the data is more important than knowing the IDs of the nodes sending it. Data aggregation is a particularly useful paradigm for wireless routing. The topology of communication ranges from simple star topology to complex multi-hop networks. The propagation technique between the hops of the network can be routing or flooding.

Cost. Cost is crucial in several applications especially because a lot of sensor nodes are deployed in most of the networks. A sensor node has several functionalities (sensing, communication, position finding, moving capability) therefore maintaining the low cost is a complex problem. Each such sensor network node has typically several parts: a sensing part, an electronic circuit for interfacing the sensors, an energy source, a radio transceiver with an internal antenna, a microcontroller, usually a battery or an embedded form of energy harvesting (The architecture of a node is shown in Fig.24.)

Reliability of the communication link. Because of the low energy resources nodes use low-power communication, which is typically not reliable. In special networks e.g. in underwater sensor networks the acoustic communication used is especially vulnerable.

Node deployment For static environment, deterministic deployment is used since the location of each sensor can be predetermined properly. The stochastic deployment is used when the information of sensing area is not known in advance or is varied with time that is the position for sensor deployment cannot be determined.

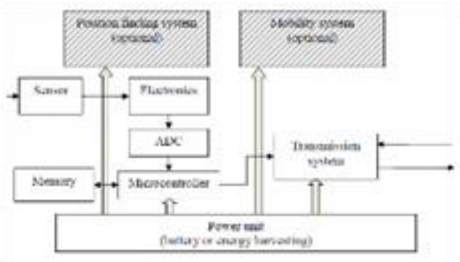
Autonomy (of sensors). In most of the networks nodes have some autonomy, which assumes some form of local intelligence. Autonomy is needed because the communication link from the sensor to central coordinator unit is vulnerable. Therefore the control of the sensor may be lost for shorter or longer periods, in that time autonomous action is needed. In case of mobile sensors autonomy is especially important.

Fault tolerance. Some sensor nodes or some communication links could be blocked due to lack of power or physical damage. In that case routing protocols have to accommodate, reroute packages.

Adaptability In several cases the nodes (the whole network) are deployed in partially unknown environment. The nodes have to adapt themselves to the environment.

Mobility (mobile sensors). In some cases the sensors are capable of changing their physical locations. It could be very important in poisonous, dangerous, unknown fields. (E.g. battlefields, fire, environmental catastrophes etc.)

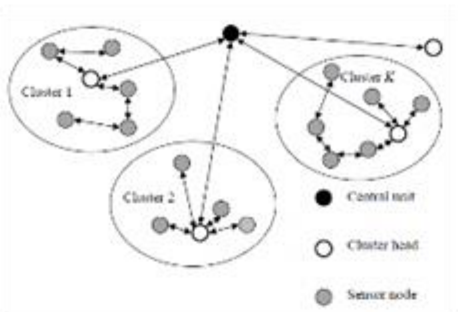
Position finding. Because in some cases the deployment could not be precisely positioned or it is even random, some nodes have to define the location where it was deployed. Similarly in mobile sensor case position finding is important.



24. ábra. Typical sensor architecture

8.2. 7.2 Typical wireless sensor network topology

A typical wireless sensor network topology is shown in Fig.25. The network consists of clusters of sensor nodes and a central unit, which collects all of the data measured by the nodes. Because communication needs relatively high energy consumption, most of the nodes pass data only to nearby nodes to a relatively short distance. In each cluster there is only one cluster head, which collects data measured by the nodes of the cluster, and communicates with the central unit of the network.



25. ábra. Typical sensor network topology

Of course this complex structure is not used in all cases; sometimes simple star topology is applied.

There are many new routing protocols designed for sensor networks. Routing protocols have several features; a recent survey gives the following aspects, and evaluates the state-of-the-art protocols (17 different protocols are surveyed) according to these ones.

The protocol could be flat, hierarchical, or location-based. About 40% of the protocols considered in the survey are flat, about 30% hierarchical, ca. 10% location-based and the remaining protocols have mixed location/hierarchical characteristic. In a hierarchical architecture nodes having higher energy are used to send the information to longer distance (especially to the central coordinator unit), while low energy nodes are used to perform the sensing task and to send the information to the closest high energy node. That means the creation of clusters and assigning special tasks to cluster heads. This architecture can greatly increase the overall system scalability. If data aggregation and fusion is performed the number of transmitted messages to the coordinator is decreased. Therefore hierarchical routing is mainly a two-layer architecture where the top layer is used to select the cluster heads, and the lower layer (within the clusters) is used for multi-hop routing.

It may apply data aggregation, which decreases the energy consumption caused by the communication needed. About 70% of the protocols surveyed have some form of aggregation.

The overhead could be low, moderate or high.

The data delivery model could be event driven, data driven, continuous, demand driven, query driven, hybrid etc. Of course this has a high impact on the energy consumption, e.g. continuous data measurement and delivery can cause unnecessary messages.

Scalability is good, limited or sometimes missing.

Quality of service - the data should be delivered within a given period of time after sensing it. Some protocols guarantee a maximum delay time, but most of the current WSN protocols do not.

8.3. 7.3 Intelligence built in sensor networks

Energy consumption must be controlled by intelligent scheduling. We have weak and - in case of harvesting - varying energy sources, so the lifetime of the sensor depends on how we expend the energy of the battery or the harvested one. The consumed power depends mainly on the active time of the sensor. Not only the consumed power but the accuracy and the delay of the data transmission depend on the sleep time - active time (duty cycle) as well. A good trade-off must be reached. The scheduling means the control of the sleep periods. If one sensor sets its own sleep period, based on self energy storage level measurement, it is called local scheduling policy, and when the central coordinator sets the sleep times for all sensors based on the entire network's state, it is called global scheduling policy. When the central coordinator manages a global policy, it can result in different sleep times for different sensors. We can use mixed policy as well, if in some cases the local policy can override the global directive.

Communication. In complex sensor networks routing is a hard and complex task, which needs local or global intelligence. Because the communication is weak frequent data losses are encountered.

Dependability. As mentioned at the paragraph dealing with energy consumption: we have weak and varying energy sources, so the lifetime of the sensor depends on how we expend the energy of the battery or the harvested one. Therefore the sensor network has to be able to solve the problem of missing sensors, if some of them fail. (The most probable cause is that it loses the energy and cannot work anymore. Of course other problems may arise as well.) Therefore the sensor network as a whole must be designed to be dependable.

Autonomy and Adaptability of the nodes requires some local and a global intelligence as well. The environment is - at least partly - unknown and changing in most of the cases.

Position finding is in most of the cases a complex and problematic task solved by intelligent algorithms. Either if mobile sensors are used, or the deployment is a random process - the establishment of the location is of crucial importance.

According to a survey of expert opinions the most important characteristics of the future sensor networks are dependability, cost and power consumption.

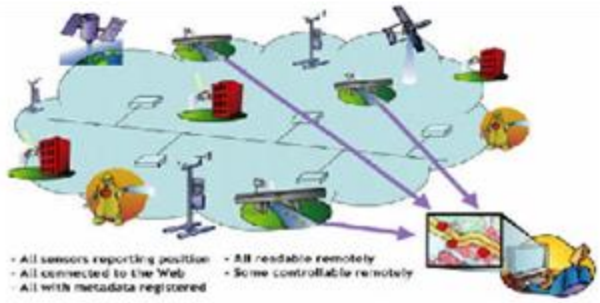
9. 8 SensorWeb. SensorML and applications

The Sensor Web is a distributed sensing system where the information is global - it is shared and used by all connected nodes. The Sensor Web can be considered a large scale measurement system composed from a number of sensor platforms. These platforms (so called pods), can be localized anywhere, can be static or mobile, if only connected.

"Sensor web" concept was coined in 1997 (by Kevin Delin, NASA) to denote a novel wireless sensor network architecture where the individual nodes could act and coordinate as a whole. Sensor web means thus a specific type of an unstructured sensor network composed of (usually) spatially distributed sensors wirelessly communicating with each other in a synchronous and router-free way.

In Delin's definition a sensor web is an autonomous, stand-alone, sensing entity - capable of interpreting and reacting to the data measured. The system is totally embedded in the monitored environment, can by itself perform data fusion acc. to the actual requirements, and reacts as a coordinated, collective whole - an intelligent agent - to the incoming data stream. Coordinated communication and interaction among the sensor nodes provides a basis for the spatial-temporal understanding of the environment (context computation), e.g. instead of having uncoordinated smoke detectors, a sensor web can react as a single, spatially dispersed, fire locator to the whole environment.

"Sensor web"' recently is associated with an additional conceptual component of having sensors connected to the World Wide Web. The Sensor Web Enablement (SWE) initiative of the Open Geospatial Consortium (OGC) defines service interfaces for an interoperable usage of sensors by enabling sensor discovery, access, programming and operating (tasking), and also define measurement based significant event information (eventing) and event based warning services (alerting). In the sensor web organized along the SWE services the heterogeneous properties of the component sensors, the communication details, are hidden from the applications. In OGC's SWE definition "sensor web" is an infrastructure enabling access to sensor networks and archived sensor data that can be discovered and accessed using standard protocols and application programming interfaces.



26. ábra. The generalized concept of the Sensor Web focuses on an architecture that is about sharing information for collective use and includes both terrestrial and orbital platforms. (G. Percival, C. Reed, **OGC® Sensor Web Enablement Standards**, Sensors and Transducers Journal, Vol.71, Issue 9, September 2006, pp.698-706)

9.1. 8.1 Sensor Web Enablement

Sensor Web Enablement means a number of objectives and tools to achieve them to implement the notion of sensor web in practical applications.

OpenGIS Consortium (http://www.opengeospatial.org/) set the following objectives:

- all sensors are connected to the Web,
- all sensors report position and observations,
- sensor are modeled and encoded in SensorML (markup language),

- access to observations made by the sensors is done through Sensor Collection Services,
- planning extensive collections of observations is done through Sensor Planning Services,
- access to sensor-related meta data is done through Web Registry Services,
- messaging is realized through Web Notification Services.

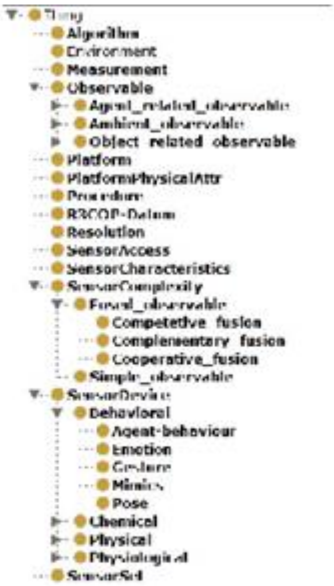
OGC® Sensor Web Enablement Standards (SWE) provide a framework for open standards for exploiting Web-connected sensors.

(http://www.opengeospatial.org/standards)

High level architecture is based on the following OGC's SWE related high level functions:

1. Discovery of sensor systems, observations, and observation (i.e. measurement and fusion) processes that fulfills an application or users requirements,
2. Identification of a sensor's capabilities and quality of measurements (i.e. observed data),
3. Access to those sensor parameters that automatically allow application software to process and geo-locate observations,
4. Retrieval of real-time or time-series observations,
5. Organizing the activity of sensors (sensor tasks) to obtain observations of interest,
6. Subscription to and publishing of alerts to be issued by sensors or sensor services (e.g. when the observable is ready).

1. Observations and Measurements (O'n M) is standard model for representing and exchanging observation results. The notions of that model are organized acc. to a sensor ontology expressing approved procedural view of the (scientific) measurement.



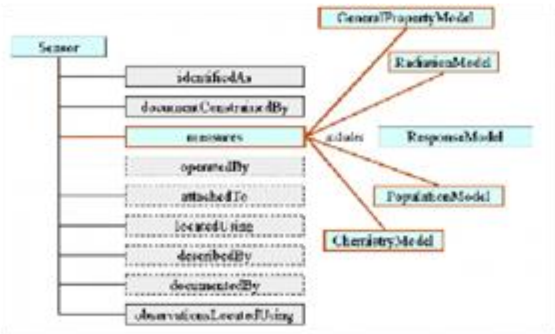
27. ábra. An example ontology showing some high level concepts related to the sensor. Sensor produces observable describing some phenomenon by an event of making measurement.

2. Sensor Model Language (SensorML) standard covers the information model and encodings, it enables discovery and tasking of Web-resident sensors, and exploitation of sensor observations. SensorML provides a functional model of the sensor system, rather than a detailed description of its hardware. In SensorML

everything, including detectors, actuators, filters, and operators is defined as process model. Process model refers to inputs, outputs, parameters, methods for that process, and collection of metadata useful for discovery and human assistance. Process metadata are identifiers, classifiers, constraints (time, legal, and security), capabilities, characteristics, contacts, and references. Information Provided by SensorML:

- Observation characteristics
  - Physical properties measured (e.g. temperature, concentration, etc.),
  - Quality characteristics (e.g. accuracy, precision),
  - Response characteristics (e.g. spectral curve, temporal response, etc.),
- Geometry characteristics
  - Size, shape, spatial weight function (e.g. point spread function) of individual samples,
  - Geometric and temporal characteristics of sample collections (e.g. scans or arrays),
- Description and documentation
  - Overall information about the sensor,
  - History and reference information supporting the SensorML document (an XML schema defining the geometric, dynamic, and observational characteristics of a sensor).
- The purpose of the sensor description is to:
  - provide general sensor information in support of data discovery,
  - support the processing and analysis of the sensor measurements,
  - support the geolocation of the measured data,
  - provide performance characteristics (e.g. accuracy, threshold, etc.),
  - archive fundamental properties and assumptions regarding sensor.

SensorML provides functional model for sensor, not detail description of hardware. Supports rigorous models, which describe sensor parameters independent of platform and target, as well as mathematical models which directly map between sensor and target space.

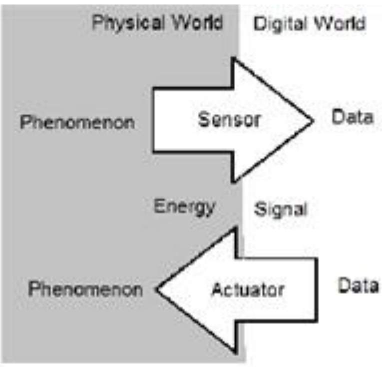


28. ábra. Hi-level diagram of SensorML showing several potential „plug-n-play” response models for the measures property. (from A Sensor Model Language: Moving Sensor Data onto the Internet, http://www.sensorsmag.com/networking-communications/a-sensor-model-language-moving-sensor-data-internet-967)

SensorML: complete description of instrument's capabilities and information needed to process and geolocate the measured data. Sensor name, type, identification numbers (identifiedAs),

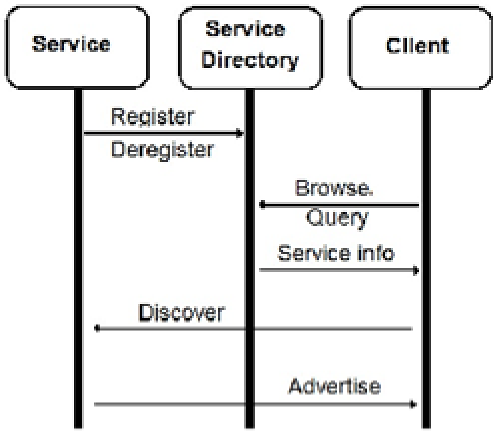
- temporal, legal, classification constraints of the description (documentConstrainedBy)
- reference to the platform description (attachedTo)
- sensor's coordinate reference system definition (hasCRS)
- sensor's location (locatedUsing)
- response characteristics and information for geolocating samples (measures)
- sensor operator and tasking services (operatedBy)
- textual metadata and history of the sensor (describedBy)
- textual metadata and history of the sensor description document itself (documentedBy).

3. TransducerML (TML) Implementation Specification models information about transducers and transducer systems capturing, exchanging, and archiving live, historical and future data received and produced by transducers (transducer: superset of sensors and actuators) Transducer is a physical or virtual entity capable of translating between physical phenomenon and transducer data. Transducers that translate phenomenon to data are called receivers or sensors. Transducers that translate data to phenomenon are called transmitters or actuators.

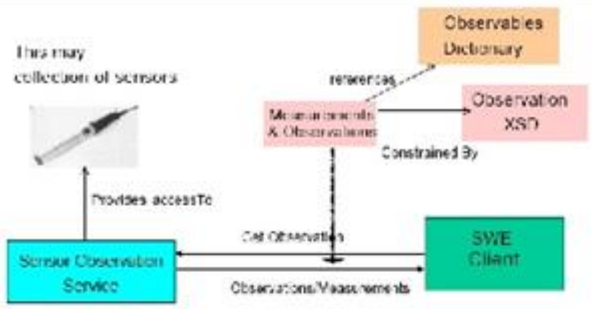


29. ábra. Conceptional difference between a transducer-sensor, and transducer-actuator (Signal is the mathematical abstraction of some energy flow).

4. Sensor Observation Service (SOS) Implementation Specification describes how to access to observations from sensors and sensor systems in a standard way that is consistent for all sensor systems including remote, in-situ, fixed and mobile sensors.



30. ábra. Using (including) a sensor in an ad hoc Sensor Web applications means „hiring" the services of that particular sensor and to this aim such services must be identified and evaluated, and this aim, in turn, the sensor must advertise somewhere its services. So „client" is the measurement oriented application, „service" is a sensor, „service directory" is a web infrastructure maintaining global service information.



31. ábra. Principal components of Sensor Observation Service. A client, expressing himself in SensorML, uses Observations and Measurement standard to obtain information about the where-about of suitable sensor platform. (G. Percival, C. Reed, **OGC® Sensor Web Enablement Standards**, Sensors and Transducers Journal, Vol.71, Issue 9, September 2006, pp.698-706)

5. Sensor Planning Service (SPS) Implementation Specification specifies interfaces for:

   - requesting information describing the capabilities of a SPS,
   - determining the feasibility of an intended sensor planning request,
   - submitting such a request,
   - inquiring about the status of such a request,
   - updating or canceling such a request,
   - requesting information about further OGC Web services (access to the data collected by the requested task).

6. Planning about sensors means building a more complex and involved measurement systems extending along the temporal and/or geographical and/or physical characteristics dimensions.

7. Sensor Alert Service (SAS) interfaces for:

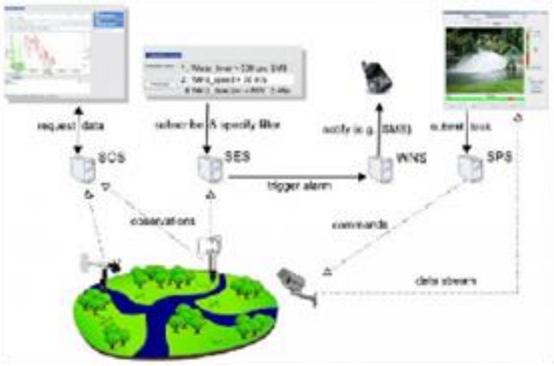
   - requesting information describing the capabilities of a Sensor Alert Service,

- determining the nature of offered alerts, the protocols, and the options to subscribe to specific alert types.

  Alert means a special kind of notification which indicate that an event has occurred at an object of interest, which results in a condition of increased watchfulness or preparation for action.

8. Web Notification Service (WNS) Interface Specification is the description of an open interface for service by which a client may conduct asynchronous dialogues (message interchanges) with one or more other services.

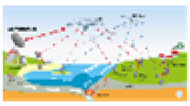
9.2. 8.2 Implementations/ applications



32. ábra. Application of SWE for hydrological deployment scenario (A. Bröring et al., New Generation Sensor Web Enablement, Sensors 2011, 11(3), 2652-2699, http://www.mdpi.com/1424-8220/11/3/2652).

Fig. 32. shows a real-world hydrological deployment scenario. The SWE services are applied to manage a network of hydrological sensors (e.g., water gauges, weather stations, or cameras observing critical facilities) by providing access to sensor data, by realizing event handling, and by enabling interoperable tasking of sensors. If a sensor web user is interested only in particular data matches some defined criteria (i.e. some feature), it can subscribe to an SES. The sensor data is continuously published to the SES and in case a specified criterion is matched, the SES forwards the data to the subscriber. Users can also register for alarms if certain events occur. In that case, the SES triggers a WNS to notify the user. E.g. a user can receive a notification via SMS or email if the water level at a gauge station is too high. Lastly, the user can utilize SPS to task sensors along more involved schedule schemes. E.g. the SPS can be used to task cameras at certain points of interest along a river (at a dam or a water gauge). The cameras can be rotated or zoomed and the real time video stream can be made available to the user.

The German Indonesian Tsunami Early-Warning System (GITEWS) (http://www.gitews.org)



33. ábra. Concept of the integrated (Sensor Web based) tsunami early warning system (http://www.gitews.org/fileadmin/images/content/Homepage/Tsunami-Zeichnung.jpg)

10. 9 Knowledge intensive information processing in intelligent spaces. Context and its components. Context management.

10.1. 9.1 Context

Schilit and Theimer (1994): context

context-aware

introducing notions.

Context (generally) = continuously changing execution (information) environment, the most important aspects are:

- where are we,
- with whom are we there,
- what resources can we count upon.

Context is any information, which can be used to characterize the situation of an entity.

An entity can be a person, place, or object, ...relevant from the point of view of the interaction between the user and the application (including the user and the application)

Context can refer to multiple environments:

- computational environment: processors, user accessible devices (input, display), network capacity, connectivity, computational costs, ...
- user environment: location, proximal humans, social situation
  - verbal context (direct communication)
    - roles of communication partners
    - aim of communication, aims of individuals
  - local environment (absolute, relative, type of environment)
  - social environment (e.g. the organization, who is there?)
- physical, chemical, biological, ...environment: lights, noise level, ...

The basic information components of a context can be characterized:

W5+ (who, where, when, what, why) questions or aspects,

A computer technical entity considers who, where, when, what aspects

to compute the why of the situation.

(designer is considering who, where, when, what, then establishes why,

and designs the behavior of the application accordingly)

(e.g. a context aware automatic museum guide:

visitor with a hand held information device steps closer to a

particular exposition, ...and the device displays the related information

(visitor, location of exposition, now, gets closer ...is interested ...to

display))

Context may be:

- primary context type: location, identity, activity, timedirect answer information: W5+index information: toward other contextual information sources
- secondary contexts (attributes of the entities in the primary context)(identity of a person ...telephone number, address, email, list of friends, ...)(localization of an entity ...who, what is around, what activities are happening around ...)

10.2. 9.2 Relevant context information

- On human:
  - static: users and visitors (e.g. identity and user profiles to control access and service personalization
  - dynamic: human actually in the environment
    - shared knowledge of mobile users about themselves: positions, movements, preferences, profiles, history of past interactions with environmental services, application specific plans, schedules, ability descriptors (for adaptive services).
    - dynamic context of mobile users:
      - location and other sensors (in user devices or environment), e.g.
      - position and path
      - actual activity, gestures, device usage patterns

    mood, emotional state"edigital' context: sessions with actual services, related events (e.g. history of navigating interface, usage of networking services, ...)
  - personal data: identity, characteristics (physical, language knowledge, personal disabilities, ...), abilities, general v. context-specific preferences and resources (usable devices).
- On locations: inclusive hierarchies (e.g. town, street, building, room, corridor)
  - physical space decomposition (space geometry)
  - logic space decomposition (application specific criteria)
  - place context: temperature, humidity, light, loudness level, ... (due to environmental sensors)
  - place: a "container" for other entities (a natural criterion to organize and analyze context) (concept of place: natural sensor fusion, transformation of a low level sensory data into semantically sound information ("situations"))
- On objects:
  - 'digital' objects, service providers for other components
  - mobile devices of the users, devices installed in the environment(household devices, PCs, printers,smart objects (physical objects equipped with sensors and computing capacity))
- On sensors:sources of dynamic information: about users, devices, applications, environment, ...
  - stand-alone devices
  - connected user wireless devices (e.g. PDAs with GPS, movement sensors, cameras, microphone, ...)
  - ambient devices
  - dynamic context: observations and events

- On situations
  - Results of fusion agents:
    - correlating user, application inputs
    - computing higher level information.
  - Situation: relating sensory data, human, objects to the spatial model of place
  - Situation ontology:
    - related to time(point)
    - related to activity making meals washing going to bed ...
    - related to health care falling forgetting taking medicine critical physiological parameters ...

10.3. 9.3 Context and information services in an intelligent space

Service entities: application servers, embedded ambient devices

(sensors, smart objects, household devices), personal (user) devices, etc.

Services:

- application-oriented,
- device- oriented
- implied aspect: access control to the service

context-awareness

context-sensitivity

A system is context aware, if to provide the user with a relevant information or service, it must use context information, where the relevancy depends on the task of the user.

10.3.1. Knowledge of context is needed for

- interpreting the communication
- interpreting other sensory information (e.g. to interpret activities)using context information:
  - in itself
  - as filter (what services are possible for the user,what interaction modalities are the best)(estimating user context - optimizing adaptation and service personalization, towards maximizing the quality of service)

10.3.2. Abilities resulting from the knowledge of context (to elevate the quality of applications)

- Contextual sensing:system is sensing the context and is passing it to the user extending the sensory apparatus of the user,
- Contextual adaptation:system is using context to adapt its behavior instead of e.g. presenting always the same interface to the user,

- Contextual resource discovery:system is able to localize and utilize context related resources,
- Contextual augmentation:system is extending the environment with further data, associating e.g. digital data with the actual context.

E.g. forwarding a message to the user may happen:

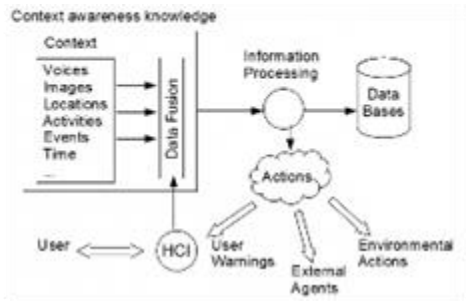
sensory information	context	context dependent adaptation
asleep	normal	storing, extracting, delivery after awaking
asleep	critical	light/sound wakening
hearing disabled	all	increased level of loudness
out-doors	normal	acknowledge, storing, delivery planned after activity
out-doors	critical	redirecting to monitoring center

4. táblázat.

10.3.3. Context dependent computations - types of applications

- presentation (information or service to the user)
- context dependent reconfiguring: provides information to the user automatically based on actual context,
- context dependent command: executing user command in a context dependent way,
- context-triggered activity: executing user command in a context dependent way (if e.g. a suitable context has been established).
- automatic execution (service)
- tagging, indexing (to compute later context information)

E.g. Searching EHR (Electronic Health Record) of the patients in a ward. If the user (medical personnel) gets closer to one of the patients, system displays his/ her automatically.



34. ábra. Context awareness.

10.4. 9.4 Context management

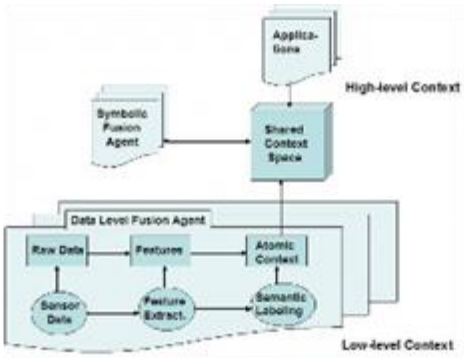
- standard protocols accessing context information
- common ontology language
- basic ontology: common entities, attributes, relations

Context query - basic problem: Pull or Push

- context query (Pull)context consumer (user, application, system, ...) actively queries the context informationconsumer controls - when asks for it and when uses itdisadvantage - query ahead in time so much that the information is fresh (valid) to use
- context mediation (Push)context providing entity notifies the potential consumerstime point of mediation is in hands of the producer (suitably often)disadvantage: who are the potential consumers? Option:
  - subscriber model
  - broadcast

  advantage/disadvantage: no need to actively query the consumertime of information arrival: anytime character, preparation - interrupt, multitasking, ...
- combined Push and Pullmediating proxy, before push, after pull (pull interface toward application, push toward the network)

10.5. 9.5 Logical architecture of context processing



35. ábra. Context spaces.

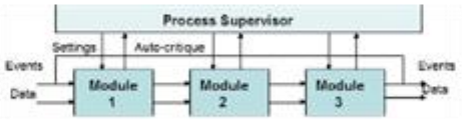
- Tasks of sensory layertransforming low level sensory data meaningful semantic context, steps:
  - obtaining rough, densely sampled sensory data
  - organizing measurement data into records, computing temporal features interval by interval (Signal Processing, Pattern Recognition)
  - feature extraction
  - binding features to human and other meaningful information (Semantic Binding)
- Tasks of semantic layer
  - publishing context data - sharable data space
  - subscription, access - applications, symbolic fusion agentssymbolic fusion agents - additional information at higher abstraction levele.g. light, humidity, temperature (sensors on the user device) - establishing situation: user out-doors, in-doors, ... signal-, feature-, pattern-, object-, behavior-, event level
- Tasks of applications:
  - access to the context space: pro-active (query, pull), publish subscribe (push)
  - discovering sources of context information (context space, format, semantics (ontologies)) similar to the Web (where is the information and in what form)

Coordinating with a shared distributed Context space distributed context space: common, distributed 'blackboard' - publish-subscribe

More advanced information management:

- handling time series beside individual observations
- mechanism of forgetting
- mechanism of historical versioning
- policy control (amount of data, refreshment scheduling, ...)
- extensions, discovering ontologies

10.6. 9.6 Feature extraction in context depending tracking of human activity



36. ábra. Context processing.

Sensing processes: series of modules, with process supervisor in control

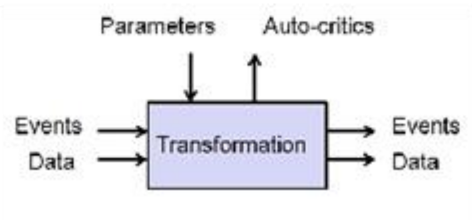
10.6.1. Process

Supervisor:

- (external) command interpretation,
- execution-scheduling, messages, scripts toward modules
- parameter adaptation, run-time reconfiguring
- reflexive description (description of state and ability for an external query)

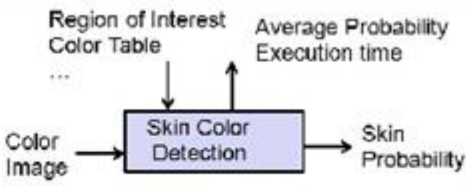
Module:

- transformation on certain data, events
- execution acc. to cyclic scheduling
- transformation: auto-critical report, result of the execution(used time, confidence, exceptions during computation, ...) input towards the supervisor for adaptation



37. ábra. Context transformation.

Basic transformations: Observation, Grouping, Tracking



38. ábra. Example: Color-observing module: pixel level observations (cheap and easy to supervise)

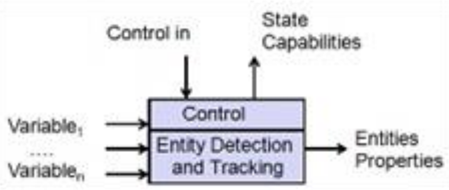
Task of tracking: prediction, observation, estimation - cyclical process, e.g. Kálmán filter

Functions of tracking:

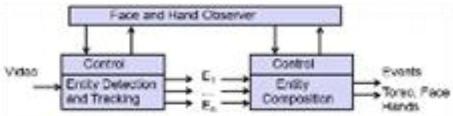
- helping interpretation by temporal integration,
- conserving information (due to recognized identity),
- focusing attention (by estimating ROI Region Of Interest),
- computing position, speed, acceleration, etc. for describing situations

Tracking/ sensing levels

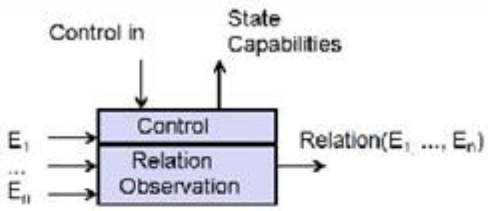
- Detecting and tracking entity
- Identifying relations between entities
- identifying groupings



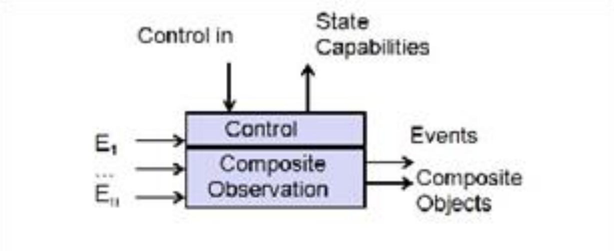
39. ábra. Composition of context computations - Entity detection.



40. ábra. Composition of context computations - Relation computation.



41. ábra. Composition of context computations - Composition observation.



42. ábra. Example of a composite computation.

Events

- Role events - change in entity-role relationE.g. in an intelligent lecture room, Lecturer changes, Camera-Aimed -At(Speaker) false, aiming (change new) camera at new person
- Situation events (Relation events)due to change in relations ...change in situationslecturer finishes the sentence and turns toward the blackboard, writes (now the camera should be directed not to face, but to the writing)
- Context eventschange in contexts, reconfiguring the sensing processes

10.7. 9.7 Example: HYCARE: context dependent reminding

Activities of Daily Living (ADLs)

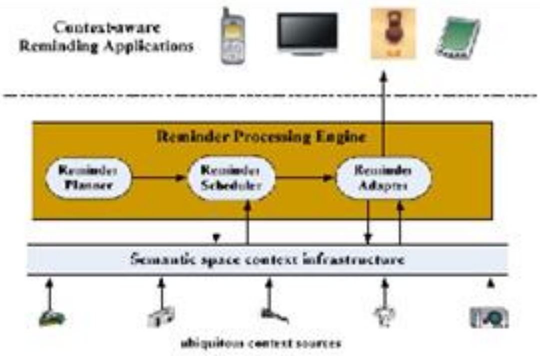
- what to do as the follow up
- hygiene
- taking medicines
- hydrating
- ...

Cognitive reinforcement:

- reminding (washing, time points, callers of the calls, forgotten objects, keys, loading up mobile, calls, cooking, taking medicines, closing the fridge, shutting down the oven, closing doors, ...)
- reinforcing social contacts
- ADL
- increased feeling of safety

10.7.1. Classification of the reminding services

- Time based prompting
  - fixed time based prompting service (definite time point, urgent notification)
  - time-relevant prompting service (relevant in time, but can be delayed within a time window)
- Event based prompting
  - prompting strictly related to an event (when the event happens)
  - event- relevant prompting



43. ábra. Processing in the reminder system. (from HYCARE: A Hybrid Context-Aware Reminding Framework for Elders with Mild Dementia, http://media.cs.tsinghua.edu.cn/ qinwj/publications/icost08.pdf)

10.7.2. Reminder Scheduler

Problems

1. There is no conflict between the reminding messages.
2. Initializing a reminding service, in the meantime activating service of higher priority.
3. Initializing a reminding service, in the meantime an external event happens, which the patient would like to handle (or the patient is not reacting to the reminders and follows with his activity)

Functioning

1. Strictly event related prompting is of higher priority.
2. Fixed time point related prompting is of second highest priority (can be interrupted or delayed only by 1.)
3. Time/Event-relevant prompting is of lowest priority

First Expired First Served (FEFS), waiting queue: ordering

Expire Time

Reminder Adapter

decides how the prompting should be directed and presented to the patient (depending upon user, device, environment) (location: a close-by device, out-doors: SMS, in-doors on TV, if watching, environment: on noisy street vibration better than ringing, ...)

11. 10 Sensor fusion.

Sensor fusion is a natural paradigm. All human beings have "sensors"; we can see, hear, taste, touch, sense temperature, some animals have even more sensors (e.g. sharks sense electric fields, some birds sense magnetic fields as well). When we interpret the current situation we perform sensor data fusion of our sensors. This biological sensor fusion system implements a heterogeneous system utilizing different types of information.

In our technical devices data fusion or sensor fusion is a relatively new field with a number of incomplete definitions. Many of these definitions are incomplete owing to its wide applicability to a number of disparate fields. We use data fusion with the narrow definition of combining the data produced by one or more sensors in a way that gives a best estimate of the quantity we are measuring.

Current data fusion ideas are dominated by two approaches or paradigms. The oldest paradigm, and the one with the strongest foundation, is Bayes theory. This theory is based on the classical ideas of probability, and has at its disposal all of the usual machinery of statistics. The Dempster-Shafer theory deals with measures of "belief" as opposed to probability. We outline the ideas of the Dempster-Shafer theory, with an example given of fusion using the cornerstone of the theory known as Dempster's rule. Dempster-Shafer theory is based on the nonclassical idea of "mass of probability" or "mass of belief" as opposed to the well-understood probabilities of Bayes theory; and although the two measures look very similar, there are some differences that we point out. We then apply the Dempster-Shafer theory to a fusion example, and point out the new ideas of "support" and "plausibility" that this theory introduces.

11.1. 10.1 Data fusion based on probability theory

Basic data fusion ideas will be shown using probabilistic representation of discrete events sampled at discrete time points. (The basic approach is the same for continuous measured variables in continuous time, but the mathematical methods are easier to use in the discrete case.)

The definition of discrete probability takes a finite or countable set called the sample space, which models the set of all possible outcomes in classical sense, denoted by Ω. It is then assumed that for each element $A_j \in \Omega$, an intrinsic "probability" value $p(A_j)$ is attached, which satisfies the following properties:

1. $p(A_j) \in [0, 1]\$$ for all $A_i \in \Omega$;
2. $\sum_{A_i \in \Omega} p(A_i) = 1$

11.1.1. Fusion of old and new data of one sensor based on Bayes-rule

First the fusion of the old and new sensory data of the same sensor is shown. Let k be the discrete time, x_k be the unknown signal to be measured, y_k be the measured value at time $k, \mathbf{Y}_k$ be the series of measurements from $1, 2, \ldots, k$. The idea is based on Bayes rule.

$$p(x_k|\mathbf{Y}_k) = \frac{p(\mathbf{Y}_k|x_k) \cdot p(x_k)}{p(\mathbf{Y}_k)} = \frac{p(y_k, \mathbf{Y}_{k-1}|x_k) \cdot p(x_k)}{p(y_k, \mathbf{Y}_{k-1})}$$

The rule could be refined if some assumptions about the system are made. The basic ideas could be understood in this simplified context as well, more complicated cases could be found in the literature. We assume that

- the measurements do not depend on previous measurements (e.g. the measurement noise is white),
- the system generating the measured signal is a Markov process, i.e. if we know the signal (x_k) at time k, then previous values are not important in predicting the future values,
- the measurements of different sensors are independent of each other.

Because of these properties the rule could be simplified:

$$p(x_k|\mathbf{Y}_k) = p(x_k|y_k, \mathbf{Y}_{k-1}) = \frac{p(y_k|x_k, \mathbf{Y}_{k-1}) \cdot p(x_k, \mathbf{Y}_{k-1})}{p(y_k, \mathbf{Y}_{k-1})} = \frac{p(y_k|x_k, \mathbf{Y}_{k-1}) \cdot p(x_k|\mathbf{Y}_{k-1}) \cdot p(\mathbf{Y}_{k-1})}{p(y_k|\mathbf{Y}_{k-1}) \cdot p(\mathbf{Y}_{k-1})} =$$
$$= \frac{p(y_k|x_k, \mathbf{Y}_{k-1}) \cdot p(x_k|\mathbf{Y}_{k-1})}{p(y_k|\mathbf{Y}_{k-1})} = \frac{p(y_k|x_k) \cdot p(x_k|\mathbf{Y}_{k-1})}{p(y_k|\mathbf{Y}_{k-1})}$$

The first term in the nominator $p(y_k|x_k)$ is the characterization of the measurement noise. The second term is a one-step prediction of the measured value, i.e. our estimate of it based on the information gathered until the previous time step. If we know the estimate of the previous value, we can easily construct this prediction using the Chapman-Kolmogorov equation:

$$p(x_k|\mathbf{Y}_{k-1}) = \sum_{x_{k-1}} p(x_k|x_{k-1}) \cdot p(x_{k-1}|\mathbf{Y}_{k-1})$$

The denominator is used to normalize the probability density, because

$$p(y_k|Y_{k-1}) = \sum_{x_k} p(y_k|x_k) \cdot p(x_k|Y_{k-1})$$

the sum of the conditional probabilities:

$$\sum_{x_k} p(x_k|Y_{k-1}) = \sum_{x_k} \frac{p(y_k|x_k) \cdot p(x_k|Y_{k-1})}{\sum_{x_k} p(y_k|x_k) \cdot p(x_k|Y_{k-1})} = 1.$$

so the sum of the probabilities is 1.

11.1.2. Fusion of data of two sensors based on Bayes-rule

The fusion of two sensors' data taken in the same time is shown. Let k be the discrete time, x_k be the unknown signal to be measured, $y_k^{(1)}$ and $y_k^{(2)}$ be the values measured by the first and second sensor at time k, $\mathbf{Y}_k^{(1)}$ and $\mathbf{Y}_k^{(2)}$ be the series of measurements of the two sensors from $1, 2, \ldots, k$. The idea is based again on Bayes rule.

$$p\left(x_k \middle| \mathbf{Y}_k^{(1)}, \mathbf{Y}_k^{(2)}\right) = p\left(x_k \middle| y_k^{(1)}, y_k^{(2)}, \mathbf{Y}_{k-1}^{(1)}, \mathbf{Y}_{k-1}^{(2)}\right) = \frac{p\left(y_k^{(1)} y_k^{(2)} \middle| x_k, Y_{k-1}^{(1)}, Y_{k-1}^{(2)}\right) p\left(x_k, Y_{k-1}^{(1)}, \right.}{p\left(y_k^{(1)}, y_k^{(2)}, \mathbf{Y}_{k-1}^{(1)}, \mathbf{Y}_{k-1}^{(2)}\right)}$$

$$p\left(x_k \middle| \mathbf{Y}_k^{(1)}, \mathbf{Y}_k^{(2)}\right) = \frac{p\left(y_k^{(1)} y_k^{(2)} \middle| x_k, \mathbf{Y}_{k-1}^{(1)}, \mathbf{Y}_{k-1}^{(2)}\right) \cdot p\left(x_k | \mathbf{Y}_{k-1}^{(1)}, \mathbf{Y}_{k-1}^{(2)}\right) \cdot p\left(\mathbf{Y}_{k-1}^{(1)}, \mathbf{Y}_{k-1}^{(2)}\right)}{p\left(y_k^{(1)}, y_k^{(2)} | \mathbf{Y}_{k-1}^{(1)}, \mathbf{Y}_{k-1}^{(2)}\right) \cdot p\left(\mathbf{Y}_{k-1}^{(1)}, \mathbf{Y}_{k-1}^{(2)}\right)} = \\ = \frac{p\left(y_k^{(1)} y_k^{(2)} \middle| x_k, \mathbf{Y}_{k-1}^{(1)}, \mathbf{Y}_{k-1}^{(2)}\right) \cdot p\left(x_k | \mathbf{Y}_{k-1}^{(1)}, \mathbf{Y}_{k-1}^{(2)}\right)}{p\left(y_k^{(1)}, y_k^{(2)} | \mathbf{Y}_{k-1}^{(1)}, \mathbf{Y}_{k-1}^{(2)}\right)}$$

The measurements $y_k^{(1)}, y_k^{(2)}$ are assumed to be independent of each other.

$$p\left(x_k \middle| \mathbf{Y}_k^{(1)}, \mathbf{Y}_k^{(2)}\right) = \frac{p\left(y_k^{(1)} \middle| x_k, \mathbf{Y}_{k-1}^{(1)}\right) \cdot p\left(y_k^{(2)} \middle| x_k, \mathbf{Y}_{k-1}^{(2)}\right) \cdot p\left(x_k | \mathbf{Y}_{k-1}^{(1)}, \mathbf{Y}_{k-1}^{(2)}\right)}{p\left(y_k^{(1)}, y_k^{(2)} | \mathbf{Y}_{k-1}^{(1)}, \mathbf{Y}_{k-1}^{(2)}\right)}$$

Using Bayes rule for the two sensors (s=1 or 2):

$$p\left(y_k^{(s)} \middle| x_k, \mathbf{Y}_{k-1}^{(s)}\right) = \frac{p(x_k | y_k^{(s)}, \mathbf{Y}_{k-1}^{(s)}) \cdot p(y_k^{(s)}, \mathbf{Y}_{k-1}^{(s)})}{p(x_k, \mathbf{Y}_{k-1}^{(s)})} = \frac{p(x_k | y_k^{(s)}, \mathbf{Y}_{k-1}^{(s)}) \cdot p(y_k^{(s)} | \mathbf{Y}_{k-1}^{(s)}) \cdot p(\mathbf{Y}_{k-1}^{(s)})}{p(x_k | \mathbf{Y}_{k-1}^{(s)}) \cdot p(\mathbf{Y}_{k-1}^{(s)})} \\ = \frac{p(x_k | y_k^{(s)}, \mathbf{Y}_{k-1}^{(s)}) \cdot p(y_k^{(s)} | \mathbf{Y}_{k-1}^{(s)})}{p(x_k | \mathbf{Y}_{k-1}^{(s)})}$$

Therefore the fusion of the two sensors:

$$p\left(x_k \middle| \mathbf{Y}_k^{(1)}, \mathbf{Y}_k^{(2)}\right) = \frac{p\left(x_k \middle| \mathbf{Y}_k^{(1)}\right) \cdot p\left(x_k \middle| \mathbf{Y}_k^{(2)}\right) \cdot p\left(x_k | \mathbf{Y}_{k-1}^{(1)}, \mathbf{Y}_{k-1}^{(2)}\right)}{p\left(x_k \middle| \mathbf{Y}_{k-1}^{(1)}\right) \cdot p\left(x_k \middle| \mathbf{Y}_{k-1}^{(2)}\right)} \cdot \frac{p(y_k^{(1)} | \mathbf{Y}_k^{(1)}) \cdot p(y_k^{(2)} |}{p(y_k^{(1)}, y_k^{(2)} | \mathbf{Y}_k^{(1)}, \mathbf{Y}}$$

The last term is simply a normalization factor.

$$p\left(x_k \middle| \mathbf{Y}_k^{(1)}, \mathbf{Y}_k^{(2)}\right) = \frac{p\left(x_k \middle| \mathbf{Y}_k^{(1)}\right) \cdot p\left(x_k \middle| \mathbf{Y}_k^{(2)}\right) \cdot p\left(x_k | \mathbf{Y}_{k-1}^{(1)}, \mathbf{Y}_{k-1}^{(2)}\right)}{p\left(x_k \middle| \mathbf{Y}_{k-1}^{(1)}\right) \cdot p\left(x_k \middle| \mathbf{Y}_{k-1}^{(2)}\right)} \cdot \frac{1}{Normalization}$$

$$p\left(x_k \middle| Y_k^{(1)}, Y_k^{(2)}\right) = \frac{p\left(x_k \middle| Y_{k-1,}^{(1)} Y_{k-1}^{(2)}\right) p\left(x_k \middle| Y_k^{(1)}\right) p\left(x_k \middle| Y_k^{(2)}\right)}{p\left(x_k \middle| Y_{k-1}^{(1)}\right) p\left(x_k \middle| Y_{k-1}^{(2)}\right)} \cdot \frac{1}{Normalization}$$

Example 10.1:

We have two different sensors one is measuring noise, the other is measuring speed. We want to identify based on these measurements the type of a vehicle coming. For sake of simplicity assume that there are only three types of vehicles: A, B, C. The measured values are roughly quantized as well, there are only 3 possible values; for the first sensor: Very Loud, Loud, Silent; for the second sensor: Very Fast, Fast, Slow. The measurement is characterized by the following tables:

Sensor 1	$x_i = A$	$x_i = B$	$x_i = C$		Sensor 2	$x_i = A$	$x_i = B$	$x_i = C$
$y_i = VeryLoud$	0.72	0.3	0.02		$y_i = VeryFast$	0.8	0.3	0.3
$y_i = Loud$	0.2	0.6	0.3		$y_i = Fast$	0.1	0.6	0.3
$y_i = Silent$	0.08	0.1	0.68		$y_i = Slow$	0.1	0.1	0.4

The sensors produce local probabilistic decisions combining their own measurements in time. Let us assume that both sensors have some a priori knowledge about the possibilities of the 3 vehicles. The first sensor takes a priori probabilities: $p(x_0 = A|Y_0^{(1)}) = 0.4;\ p(x_0 = B|Y_0^{(1)}) = 0.4;\ p(x_0 = C|Y_0^{(1)}) = 0.2$, the second one takes that: : $p(x_0 = A|Y_0^{(2)}) = 0.6;\ p(x_0 = B|Y_0^{(2)}) = 0.2;\ p(x_0 = C|Y_0^{(2)}) = 0.1$. What will be the probabilities assigned to the vehicle types at $k = 1$, if the first sensor measures VeryLoud, the second one measures VeryFast at $k = 1$. The following equation will be used:

$$p(x_1|\mathbf{Y}_1) = \frac{p(y_1|x_1) \cdot \sum_{x_0} p(x_1|x_0) \cdot p(x_0|\mathbf{Y}_0^{(1)})}{p(y_1|\mathbf{Y}_0^{(1)})}$$

and it is utilized that the probability transitions are special in that case, the type of the vehicle coming cannot change through time (x_{k-1} must be the same as x_k). For example:

$$p(x_1 = A|\mathbf{Y}_1^{(1)}) = \frac{p(y_1 = V1|x_1 = A) \cdot \sum_{x_0} p(x_1 = A|x_0) \cdot p(x_0|\mathbf{Y}_0^{(1)})}{Norm} =$$
$$= \frac{p(y_1 = V1|x_1 = A) \cdot \left[p(x_1 = A|x_0 = A) \cdot p(x_0 = A|\mathbf{Y}_0^{(1)})\right]}{Norm} = \frac{0.72 \cdot [1 \cdot 0.4]}{Norm} = \frac{0.288}{Norm}$$

After similar computations:

$$p(x_1 = B|\mathbf{Y}_1^{(1)}) = \frac{p(y_1 = V1|x_1 = B) \cdot \left[p(x_1 = B|x_0 = B) \cdot p(x_0 = B|\mathbf{Y}_0^{(1)})\right]}{Norm} = \frac{0.3 \cdot [1 \cdot 0}{Norm}$$

$$p(x_1 = C|\mathbf{Y}_1^{(1)}) = \frac{p(y_1 = V1|x_1 = C) \cdot \left[p(x_1 = C|x_0 = C) \cdot p(x_0 = C|\mathbf{Y}_0^{(1)})\right]}{Norm} = \frac{0.02 \cdot [1 \cdot (}{Norm}$$

Because the sum of the three conditional probabilities must be 1, therefore $Norm = 0.288 + 0.12 + 0.004 = 0.412$. The three probabilities using the normalization

$$p(x_1 = A|\mathbf{Y}_1^{(1)}) = \frac{0.288}{Norm} = \frac{0.288}{0.412} = 0.7$$

$$p(x_1 = B|\mathbf{Y}_1^{(1)}) = \frac{0.12}{Norm} = \frac{0.12}{0.412} = 0.29$$

$$p(x_1 = C|\mathbf{Y}_1^{(1)}) = \frac{0.004}{Norm} = \frac{0.004}{0.412} = 0.01$$

After similar computations the probabilities given by the second sensor:

$$p(x_1 = A|\mathbf{Y}_1^{(2)}) = \frac{0.48}{Norm} = \frac{0.48}{0.6} = 0.8$$

$$p(x_1 = B|\mathbf{Y}_1^{(2)}) = \frac{0.09}{Norm} = \frac{0.09}{0.6} = 0.15$$

$$p(x_1 = C|\mathbf{Y}_1^{(2)}) = \frac{0.03}{Norm} = \frac{0.03}{0.6} = 0.05$$

11.1.3. Sensor data fusion based on Kalman filtering

Basically linear Kalman filtering assumes that we know the structure and the parameters of the unknown system. The equation describing the system's behavior:

$$\mathbf{x_{k+1}} = \mathbf{F} \cdot \mathbf{x_k} + \mathbf{B} \cdot \mathbf{u_{k+1}} + \mathbf{v_{k+1}}$$

Here k is the discrete time, $\mathbf{x_k}$ is the state vector of the system, $\mathbf{u_k}$ is the input vector at time , $\mathbf{B}$ is the input matrix; and $\mathbf{v_k}$ is a noise modeling the uncertainty of our knowledge of the next state of the system. We know $\mathbf{F}$, $\mathbf{B}$; and the covariance matrix of the noise: $\mathbf{Q}$. The measurement is described by the equation:

$$\mathbf{y_k} = \mathbf{H} \cdot \mathbf{x_k} + \mathbf{w_k}$$

Where $\mathbf{y_k}$ is the vector of the measured values, $\mathbf{H}$ is the measurement matrix; and $\mathbf{v_k}$ is the measurement noise. The covariance of the measurement noise is known, $\mathbf{R}$. (The parameters of the system, the system noise and measurement noise could depend on the time as well, but for the sake simplicity we take stationer situation.)

Kalman filter gives an optimal estimate of the state vector, and the variance of the state vector. If the estimate at time $k+1$ is based on the information gathered until the previous time instant (k), we call it one step prediction and it is denoted by $\widehat{\mathbf{x}}(k+1|k)$ and $\widehat{\mathbf{P}}(k+1|k)$ respectively. The Kalman filter algorithm uses the following recursive method:

1. We start the algorithm at $k = 0$ time. Initial a priori estimates of $\widehat{\mathbf{x}}(0|0)$ and $\widehat{\mathbf{P}}(0|0)$ are given.
2. A prediction of the next state vector is given, using the knowledge of the system and the input:
3. $\hat{\mathbf{x}}(k+1|k) = \mathbf{F} \cdot \hat{\mathbf{x}}(k|k) + \mathbf{B} \cdot u_{k+1}$
4. The variance of the predicted state vector is estimated:

5. $\hat{\mathbf{P}}(k+1|k) = \mathbf{F} \cdot \hat{\mathbf{P}}(k|k)\mathbf{F}^T + \mathbf{Q}$

6. The Kalman gain is evaluated:

7. $\mathbf{K}(k+1) = \hat{\mathbf{P}}(k+1|k) \cdot \mathbf{H}^T \cdot \left[\mathbf{H} \cdot \hat{\mathbf{P}}(k+1|k) \cdot \mathbf{H}^T + \mathbf{R}\right]^{-1}$

8. The new measurement is predicted based on the predicted state vector and the measurement equation. A real measurement is taken, and the new real measurement is compared to the predicted measurement values:

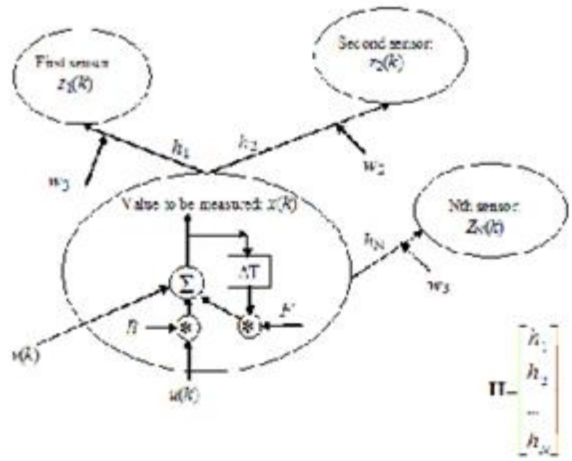
9. $\hat{\mathbf{z}}(k+1) = \mathbf{H} \cdot \hat{\mathbf{x}}(k+1|k)\, \hat{\mathbf{x}}(k+1|k+1) = \hat{\mathbf{x}}(k+1|k) + \mathbf{K}(k+1) \cdot [\mathbf{z}(k+1) - \hat{\mathbf{z}}(k+1)]$

10. The variance of the estimated state vector is given:

11. $\hat{\mathbf{P}}(k+1|k+1) = [1 - \mathbf{K}(k+1) \cdot \mathbf{H}] \cdot \hat{\mathbf{P}}(k+1|k)$

12. The next time step is taken: $k \leftarrow k+1$; and the algorithm continues at step 2.

The use of Kalman filter as a sensor fusion algorithm is straightforward. In Fig 44 the measurement of a scalar x signal using N sensors is shown. The unknown parameter to be measured is modeled by the linear system equation. If the signal is changing fast, then F is of a high absolute value (either positive if the signal is increasing fast, or negative if it is decreasing fast.) If the system is deterministic or nearly deterministic, then Q is a small value, the system noise is relatively low. If the change of the parameter to be measured is very unpredictable, then Q is a high value.



44. ábra. Operation of Kalman filter.

The Kalman filter shown above gives a fusion algorithm by combining the measured values of the individual sensors in an optimal way.

Example 10.2:

We measure the temperature of a room in every hour using 3 sensors. The room is heated but the heating causes a slow temperature increase (0.1^0 C/hour) only. We model the heating by using $\mathbf{F} = 1,\ \mathbf{B} = 0.1\ u_k = 1,\ k = 0, 1, 2, \ldots$. There are uncertainties in the heating process (sometimes the doors and windows are opened, the sunlight gives some extra heating etc.); and there is noise in the measurements. The uncertainty of the heating is characterized by a white noise process with Q=0.01 covariance. The measurement is modeled by:

$$\mathbf{H}^T = [1, 1, 1] \text{ and noise covariance matrix } \mathbf{R} = \begin{bmatrix} 0.8 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Our initial estimates are $\hat{x}(0|0) = 0$ and $\hat{P}(0|0) = 0$. Let us compute the estimated value of the room temperature at $k=1$, if our measured values are:

$$\mathbf{z}(1) = \begin{bmatrix} 0.243 \\ 0.0659 \\ 0.8291 \end{bmatrix}$$

The prediction of the temperature based on the initial values and the knowledge of the system:

$$\hat{\mathbf{x}}(1|0) = 1 \cdot \hat{\mathbf{x}}(0|0) + \mathbf{B} \cdot u_1 = 1 \cdot 0 + 1 \cdot 0.1 = 0.1$$

The variance of the predicted state vector is estimated:

$$\hat{\mathbf{P}}(1|0) = \mathbf{F} \cdot \hat{\mathbf{P}}(k|k) \cdot \mathbf{F}^T + \mathbf{Q} = 1 \cdot 0 \cdot 1 + 0.01 = 0.01$$

The Kalman gain is computed:

$$\mathbf{K}(1) = \hat{\mathbf{P}}(1|0) \cdot \mathbf{H}^T \cdot \left[\mathbf{H} \cdot \hat{\mathbf{P}}(1|0) \cdot \mathbf{H}^T + \mathbf{R}\right]^{-1} = 0.01 \cdot \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \cdot \left[\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \cdot 0.01 \cdot \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}\right.$$

$$\mathbf{K}(1) = \begin{bmatrix} 0.0117 & 0.0466 & 0.0093 \end{bmatrix}$$

The estimated measurement values are compared to the real measurements:

$$\hat{\mathbf{x}}(1|1) = \hat{\mathbf{x}}(1|0) + \mathbf{K}(1) \cdot [\mathbf{z}(1) - \mathbf{H} \cdot \hat{x}(1|0)] =$$
$$= 0.1 + \begin{bmatrix} 0.0117 & 0.0466 & 0.0093 \end{bmatrix} \cdot \left[\begin{bmatrix} 0.243 \\ 0.0659 \\ 0.8291 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \cdot 0.1\right] = 0.1069$$

The uncertainty of the estimation could be estimated as well:

$$\hat{\mathbf{P}}(1|1) = [1 - \mathbf{K}(1) \cdot \mathbf{H}] \cdot \hat{\mathbf{P}}(1|0) = \left[1 - \begin{bmatrix} 0.0117 & 0.0466 & 0.0093 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}\right] \cdot 0.1 = 0.0093$$

Remarks:

1. Accidentally in our example $\widehat{\mathbf{P}}(1|1)$ is equal to the third component of $\mathbf{K}(1)$, but it is not a rule, in other situations it will not be true.

2. The resulted Kalman gain - $\mathbf{K}(1)$ - is quite reasonable, because the second component of it is weighting the second sensor's measurement, and this weighting component is the largest one. It is reasonable because the second sensor have the smallest noise (variance is 0.2). Similarly the worst measurement has the lowest weight; the third measurement is in the middle.

11.2. 10.2 Dempster-Shafer theory of fusion

Dempster-Shafer theory deals with the uncertainty of the data measured in a different way compared to the classical probability theory based Bayes fusion. The Dempster-Shafer theory deals with measures of "mass of probability" or "mass of belief" as opposed to probability. The most important new idea in this approach is that our knowledge and our beliefs are to be modeled, inclusive our ignorance as well. The events used to model the situation are not necessarily distinct; and events could be at different granularity levels; especially the complex event modeling the ignorance is composed from all the possible outcomes.

As mentioned the definition of discrete probability takes a finite or countable set called the sample space, which models to the set of all possible outcomes, denoted by Ω. It is then assumed that for each element $A_i \in \Omega$, a "probability" value $p(A)$ is attached, which satisfies the following properties:

$p(A_i) \in [0,1]\$$ for all $A_i \in \Omega$;

$\sum_{A_i \in \Omega} p(A_i) = 1$

It should be emphasized that all the elementary events or outcomes are distinct. On the other hand Dempster-Shafer theory models these uncertain events typically with both simple and composite events, which are not necessarily distinct.

Example 10.3:

We want to detect vehicles. We characterize the vehicle detected by using the categories like motorbikes, cars, trucks, buses, pickups. But we can have a complex category like BigVehicles={buses,trucks}, or the total ignorance: WeDontKnow={motorbikes, cars, trucks, buses, pickups}.

Example 10.4:

E10.41: Let the basic distinct uncertain events are {A, B, C, D, E}. In the probability theory some values e.g. p(A)=0.13; p(B)=0.21; p(C)=0.5; p(D)=0.09; p(E)=0.07 characterize the possibility that the given outcome will happen. All these values are in the $[0,1]$; and the sum of them gives 1.

Dempster-Shafer theory models our beliefs in a different way. Some elementary or complex events are modeled, e.g. the possible outcomes could be: {{A},{B},{C},{D},{E},{B,C},{A,B,C,D,E}}. The last one of the set is the total ignorance: we do not know which of the events could occur. We assign values to al the events in the $[0,1]$ range characterizing the uncertainty, the sum is again 1. They are clearly different from the probability values, in the Dempster-Shafer theory it is called "mass of probability". E.g. m({A})=0.1; m({B})=0.12; m({C})=0.07; m({D})=0.05; m({E})=0.17; m({B,C})=0.19; m({A,B,C,D,E})= 0.3.

E10.42: If we want to give some meaning to the example E10.41, we can think of the vehicle recognition problem based on the sound we detect.

In the probability theory Bayes rule is used to combine probabilities. The basic question what to do with the masses of probability assigned to the events in the Dempster-Shafer scenario if we want to combine two pieces of information. The original suggestion was the same combination rule both if we combine old a new data of the same sensor; or we combine current data of two different sensors.

Fusion of old and new data (mass of probability) of a given sensor:

$$m(C) = \frac{\sum_{A \cap B = C} m_{old}(A) \cdot m_{new}(B)}{1 - \sum_{A \cap B = 0} m_{old}(A) \cdot m_{new}(B)}$$

Data fusion of two different sensors (sensor1 and sensor2):

$$m^{(1,2)}(C) = \frac{\sum_{A \cap B = C} m^{(1)}(A) \cdot m^{(2)}(B)}{1 - \sum_{A \cap B = 0} m^{(1)}(A) \cdot m^{(2)}(B)}$$

The strength of Dempster-Shafer fusion could be best understood by analyzing an example.

We detect vehicles and there are 3 types: A, B and C. Two of them are fast ones and C is relatively slow. We have a sensor which assigns some mass of probability (MOP) to the categories modeled. (It is a Dempster-Shafer version of Example 10.1.)

At time k the sensor assigns the following MOPs to the categories:

vehicle	{A}	{B}	{C}	Fast={A, B}	Θ=Ignorance={A, B, C}
MOP(k)	0.3	0.2	0.1	0.3	0.1
MOP(k+1)	0.4	0.3	0.2	0	0.1

5. táblázat.

Let us see the combined MOP of category {A}, if we combine the two information using the Dempster rule:

$$m(\{A\}) = \frac{\sum\limits_{X \cap Y = \{A\}} m_{old}(X) \cdot m_{new}(Y)}{1 - \sum\limits_{X \cap Y = 0} m_{old}(X) \cdot m_{new}(Y)}$$

The nominator reflects all the combinations which intersect in {A}

$$\text{Nominator} = \sum_{X \cap Y = \{A\}} m_{old}(X) \cdot m_{new}(Y) = m_{old}(\{A\}) \cdot m_{new}(\{A\}) + m_{old}(\{A\}) \cdot m_{new}$$
$$+ m_{old}(Fast) \cdot m_{new}(\{A\}) + m_{old}(\Theta) \cdot m_{new}(\{A\}) =$$
$$= 0.3 \cdot 0.4 + 0.3 \cdot 0.3 + 0.3 \cdot 0.1 + 0.3 \cdot 0.4 + 0.3 \cdot 0.1 + 0.1 \cdot 0.4 = 0.31$$

The denominator is used to normalize all the relevant MOPs, therefore the sum of them will be 1. Note that only events resulted from contradiction are excluded (e.g. both L and F cannot be true together):

$$\text{Denom} = 1 - m_{old}(\{A\}) \cdot m_{new}(\{B\}) - m_{old}(\{A\}) \cdot m_{new}(\{C\}) -$$
$$- m_{old}(\{B\}) \cdot m_{new}(\{C\}) - m_{old}(\{B\}) \cdot m_{new}(\{A\}) - m_{old}(\{C\}) \cdot m_{new}(\{A\}) -$$
$$- m_{old}(\{C\}) \cdot m_{new}(\{B\}) - m_{old}(\{C\}) \cdot m_{new}(Fast) - m_{old}(Fast) \cdot m_{new}(\{C\})$$

$$\text{Denom} = 1 - 0.3 \cdot 0.3 - 0.3 \cdot 0.2 - 0.2 \cdot 0.2 - 0.2 \cdot 0.4 - 0.1 \cdot 0.4 - 0.1 \cdot 0.3 - 0.1 \cdot 0 - 0.3 \cdot 0.2 = 0.6$$

The combined mass of probability assigned to {A}:

$$m^{(old \otimes new)}(\{A\}) = \frac{0.31}{0.6} = 0.52$$

Let us show the MOP assigned to the ignorance for comparison:

$$\text{Nominator} = \sum_{X \cap Y = \Theta} m_{old}(X) \cdot m_{new}(Y) = m_{old}(\Theta) \cdot m_{new}(\Theta) = 0.1 \cdot 0.1 = 0.01$$

The denominator is the same, so:

$$m^{(old \otimes new)}(\Theta) = \frac{0.01}{0.6} = 0.017$$

The method is similar if we combine the MOPs of two sensors. If the MOPs assigned by the two sensors are the same as the previous case, the steps and the result will be the same.

vehicle	{A}	{B}	{C}	Fast={A, B}	Θ=Ignorance={A, B, C}
MOP$^{(1)}$	0.3	0.2	0.1	0.3	0.1
MOP$^{(2)}$	0.4	0.3	0.2	0	0.1

6. táblázat.

$$m^{(1\otimes 2)}(\{A\}) = \frac{0.31}{0.6} = 0.52$$

11.2.1. Sensors of different reliability

If some information of the sensors' reliability is known, it is reasonable to weight the data provided by different sensors. The more reliable the sensor is, the more we trust the data provided by it. A reasonable suggestion to weigh the MOPs is the following.

Let us assume that the k th sensor has a weight of w_k, and it provides MOPs for the events modeled. In the fusion process the following modified MOPs are to be used:

$$m^{(k)*}(A) = w_k \cdot m^{(k)*}(A) \qquad \text{for all } A \subset \Theta \text{ (but! } A \neq \Theta)$$

$$m^{(k)*}(\Theta) = w_k \cdot m^{(k)*}(\Theta) + 1 - w_k$$

Note that for $w_k = 1$ the MOPs assigned are not changed.

Example 10.5:

In our system there are two sensors, the fusion of their data is used. Three events (A, B; and $\Theta = \{A, B\}$) are modeled using Dempster-Shafer theory. The two sensors assign the following MOPs to the events:

$$\begin{array}{l} m^{(1)}(A) = 0.6 m^{(2)}(A) = 0.2 \\ m^{(1)}(B) = 0.2 m^{(2)}(A) = 0.6 \\ m^{(1)}(\Theta) = 0.2 m^{(2)}(\Theta) = 0.2 \end{array}$$

If both sensors are equally reliable ($w_1 = 1, w_2 = 1$), the fusion result is:

$$\begin{array}{l} q^{(1\otimes 2)}(A) = m^{(1)}(A) \cdot m^{(2)}(A) + m^{(1)}(A) \cdot m^{(2)}(\Theta) + m^{(1)}(\Theta) \cdot m^{(2)}(A) = 0.28 \\ q^{(1\otimes 2)}(B) = m^{(1)}(B) \cdot m^{(2)}(B) + m^{(1)}(B) \cdot m^{(2)}(\Theta) + m^{(1)}(\Theta) \cdot m^{(2)}(B) = 0.28 \\ q^{(1\otimes 2)}(\Theta) = m^{(1)}(\Theta) \cdot m^{(2)}(\Theta) = 0.04 \\ q^{(1\otimes 2)}(0) = 1 - \sum\limits_{X \cap Y = 0} m^{(1)}(X) \cdot m^{(2)}(Y) = 1 - (m^{(1)}(A) \cdot m^{(2)}(B) + m^{(1)}(B) \cdot m^{(2)}(A)) = \end{array}$$

$$\begin{array}{l} m^{(1\otimes 2)}(A) = \frac{q^{(1\otimes 2)}(A)}{1 - q^{(1\otimes 2)}(0)} = \frac{0.28}{0.6} = 0.47 \\ m^{(1\otimes 2)}(B) = \frac{q^{(1\otimes 2)}(B)}{1 - q^{(1\otimes 2)}(0)} = \frac{0.28}{0.6} = 0.47 \\ m^{(1\otimes 2)}(\Theta) = \frac{q^{(1\otimes 2)}(\Theta)}{1 - q^{(1\otimes 2)}(0)} = \frac{0.04}{0.6} = 0.067 \end{array}$$

If the second sensor is not really reliable ($w_1 = 1, w_2 = 0.1$), the modified MOPs are:

$$\begin{array}{l} m^{(1)*}(A) = 0.6 m^{(2)*}(A) = 0.02 \\ m^{(1)*}(B) = 0.2 m^{(2)*}(A) = 0.06 \\ m^{(1)*}(\Theta) = 0.2 m^{(2)*}(\Theta) = 0.92 \end{array}$$

The fusion result is:

$$q^{(1\otimes 2)}(A) = m^{(1)*}(A) \cdot m^{(2)*}(A) + m^{(1)*}(A) \cdot m^{(2)*}(\Theta) + m^{(1)*}(\Theta) \cdot m^{(2)*}(A) = 0.568$$
$$q^{(1\otimes 2)}(B) = m^{(1)*}(B) \cdot m^{(2)*}(B) + m^{(1)*}(B) \cdot m^{(2)*}(\Theta) + m^{(1)*}(\Theta) \cdot m^{(2)*}(B) = 0.208$$
$$q^{(1\otimes 2)}(\Theta) = m^{(1)*}(\Theta) \cdot m^{(2)*}(\Theta) = 0.184$$
$$1 - q^{(1\otimes 2)}(0) = 1 - \sum_{X \cap Y = 0} m^{(1)}(X) \cdot m^{(2)}(Y) = 1 - (m^{(1)*}(A) \cdot m^{(2)*}(B) + m^{(1)*}(B) \cdot m^{(2)}$$

$$m^{(1\otimes 2)*}(A) = \frac{q^{(1\otimes 2)*}(A)}{1-q^{(1\otimes 2)*}(0)} = \frac{0.568}{0.96} = 0.592$$
$$m^{(1\otimes 2)*}(B) = \frac{q^{(1\otimes 2)*}(B)}{1-q^{(1\otimes 2)*}(0)} = \frac{0.208}{0.96} = 0.217$$
$$m^{(1\otimes 2)*}(\Theta) = \frac{q^{(1\otimes 2)*}(\Theta)}{1-q^{(1\otimes 2)*}(0)} = \frac{0.184}{0.96} = 0.192$$

This result is quite rational, the MOPs of the better sensor have greater effect on the final (fusioned) results - e.g. on $m^{(1\otimes 2)^*}(A)$ - than the MOPs of the worse one.

There is a basic problem in the original Dempster-Shafer theory. If the conflict of two sensors (contradiction) is important the result will be unreasonable. The well known example showing the phenomenon is the case of two sensors (or experts) and three events (A, B, C). (In the original example there are two physicians, expert1 and expert2; and 3 illnesses A, B and C.)

event	A	B	C	Θ=Ignorance={A, B, C}
MOP$^{(1)}$	0.99	0.01	0	0
MOP$^{(2)}$	0	0.01	0.99	0

7. táblázat.

The combination of the MOPs of sensor1 and sensor2 will result the following:

$$m^{(1\otimes 2)}(A) = \frac{0}{1-0.9999} = 0; m^{(1\otimes 2)}(B) = \frac{0.0001}{1-0.9999} = 1;$$
$$m^{(1\otimes 2)}(C) = \frac{0}{1-0.9999} = 0; m^{(1\otimes 2)}(\Theta) = \frac{0}{1-0.9999} = 0$$

The combined information assigns 1 (100 %) mass of probability to event B, which was not really trusted by any of the sensors (experts). There are two causes of that phenomenon. First the sensors did not assign any MOP to the ignorance (these experts are a bit too self-confident). Second the Dempster-Shafer rule excludes all the conflicts.

Let us see what happens if the two experts (sensors) give at least some MOP to ignorance.

event	A	B	C	Θ=Ignorance={A, B, C}
MOP$^{(1)}$	0.98	0.01	0	0.01
MOP$^{(2)}$	0	0.01	0.98	0.01

8. táblázat.

The new combined MOPs:

$$m^{(1\otimes 2)}(A) = \frac{0.0098}{1-0.98} = 0.49; m^{(1\otimes 2)}(B) = \frac{0.0001}{1-0.98} = 0.005;$$
$$m^{(1\otimes 2)}(C) = \frac{0.0098}{1-0.98} = 0.49; m^{(1\otimes 2)}(\Theta) = \frac{0.0001}{1-0.98} = 0.005$$

These results are reasonable, the two events A and C, which were believable at least for one of to the two sensors have nearly 50% MOP, and the third event (not really probable for any of the sensors) and the ignorance both have some MOP as well.

11.2.2. Yager's combination rule

Yager gave a new idea how to deal with conflicts. It was indicated that normalization (the denominator of the Dempster-Shafer combination rule) causes the basic problem dealing with conflicts. According to the new suggestion the nominator of the rule is kept, and the resulting mass of belief is called "ground mass of probability" (gMOP).

$q^{(1\otimes 2)}(C) = \sum_{A\cap B=C} q^{(1)}(A)\cdot q^{(2)}(B)$ for $\forall A$, $A \subseteq \Theta$

The new idea is, that the conflict is assigned to a gMOP

$$q(0) = \sum_{A\cap B=0} q^{(1)}(A)\cdot q^{(2)}(B) \geq 0$$

This gMOP of the conflict increases the gMOP of the ignorance:

$$q(\Theta) = q(\Theta) + q(0)$$

Let us see the effect of the new combination rule on the previous example.

event	A	B	C	0,conflict	Θ=Ignorance={A, B, C}
gMOP$^{(1)}$	0.98	0.01	0		0
gMOP$^{(2)}$	0	0.01	0.99		0
gMOP$^{(1\oplus 2)}$	0	0.0001	0	0.9999 ⇒	0.9999

9. táblázat.

This result is a more reasonable one, because of the sharp conflict, the combined information shows high level of ignorance. The event B has some very small gMOP, which is consistent with the small beliefs assigned to this event by both of the sensors/experts.

11.2.3. Inakagi's unified combination rule

It was shown [Wu2003B] that both Dempster-Shafer and Yager rules are special cases of a general, unified combination rule. Inakagi's rule keeps the ground mass of probability concept of the Yager theory, and the nonzero gMOP assigned to the conflicts.

$q^{(1\otimes 2)}(C) = \sum_{A\cap B=C} q^{(1)}(A)\cdot q^{(2)}(B)$ for $\forall A$, $A \subseteq \Theta$

$$q^{(1\oplus 2)}(0) = \sum_{A\cap B=0} q^{(1)}(A)\cdot q^{(2)}(B) \geq 0$$

It uses a k parameter, which defines the special rules derived from the general, unified one.

$m_U^{(1\oplus 2)}(A) = \left[1 + k\cdot q^{(1\oplus 2)}(0)\right]\cdot q^{(1\oplus 2)}(A)$ for all A , $A \neq 0$ és $A \neq \Theta$

$$m_U^{(1\oplus 2)}(\Theta) = \left[1 + k\cdot q^{(1\oplus 2)}(0)\right]\cdot q^{(1\oplus 2)}(\Theta) + \left[1 + k\cdot q^{(1\oplus 2)}(0) - k\right]\cdot q^{(1\oplus 2)}(0)$$

$$0 \leq k \leq \frac{1}{1 - q(0) - q(\Theta)}$$

It should be noted that if the gMOP values of the sensors sum up to 1 each, then the unified rule will produce probability masses, which sum up to 1 again. Formally

$S^{(1)} = \sum_{A_j\neq\Theta} q^{(1)}(A_j) + q^{(1)}(\Theta) = 1$,

$$S^{(2)} = \sum_{A_k \neq \Theta} q^{(2)}(A_k) + q^{(2)}(\Theta) = 1$$

Multiplying these two sums the combined ground probabilities of every event (the ignorance included) plus the gMOP of the conflict will be given. Therefore the combined gMOPs will sum up to 1:

$$S^{(1\oplus 2)} = S^{(1)} \cdot S^{(2)} = \sum_{A_j \neq \Theta} q^{(1\oplus 2)}(A_j) + q^{(1\oplus 2)}(\Theta) + q^{(1\oplus 2)}(0) = 1$$

Therefore $\sum_{A_j \neq \Theta} q^{(1\oplus 2)}(A_j) + q^{(1\oplus 2)}(\Theta) = 1 - q^{(1\oplus 2)}(0)$.

$$m_U^{(1\oplus 2)}(\Theta) + \sum_{A_k \neq \Theta} m_U^{(1\oplus 2)}(A_k) = \left[1 + k \cdot q^{(1\oplus 2)}(0)\right] \cdot q^{(1\oplus 2)}(\Theta) + \left[1 + k \cdot q^{(1\oplus 2)}(0) - k\right] \cdot$$
$$+ \sum_{A_k \neq \Theta} \left[1 + k \cdot q^{(1\oplus 2)}(0)\right] \cdot q^{(1\oplus 2)}(A_k)$$

$$m_U^{(1\oplus 2)}(\Theta) + \sum_{A_k \neq \Theta} m_U^{(1\oplus 2)}(A_k) = \left[1 + k \cdot q^{(1\oplus 2)}(0)\right] \cdot \sum_{\substack{A_k \neq \Theta \\ A_k = \Theta}} \cdot q^{(1\oplus 2)}(A_k) + \left[1 + k \cdot q^{(1\oplus 2)}(0) - k\right.$$

$$m_U^{(1\oplus 2)}(\Theta) + \sum_{A_k \neq \Theta} m_U^{(1\oplus 2)}(A_k) = \left[1 + k \cdot q^{(1\oplus 2)}(0)\right] \cdot \left[1 - q^{(1\oplus 2)}(0)\right] + \left[1 + k \cdot q^{(1\oplus 2)}(0)\right] \cdot q^{(1\oplus}$$

It is obvious that if $k = 0$ the above equations give the Yager rule. If $k = \frac{1}{1 - q^{(1\oplus 2)}(0)}$ the original Dempster-Shafer rule is resulted.

It is worth to analyze how $m_U^{(1\oplus 2)}(\Theta)$ - mass of probability assigned to ignorance - depends on k.

k	0	$\frac{1}{1-q(0)}$	$\frac{1}{1-q(0}$
$m_U^{(1\oplus 2)}(\Theta)$	$q^{(1\oplus 2)}(\Theta) + \cdot q^{(1\oplus 2)}(0)$	$q^{(1\oplus 2)}(\Theta) \leq m_U^{(1\oplus 2)} \leq q^{(1\oplus 2)}(\Theta) + \cdot q^{(1\oplus 2)}(0)$	$q^{(1\oplus}$

10. táblázat.

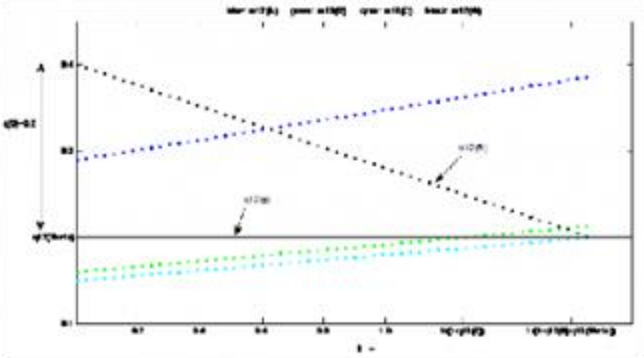
At $k = 0$ the gMOP of the conflict is added to the gMOP of the ignorance. At the other border of the k interval only the gMOP of the ignorance gives the final MOP of it. In that case as we will see, the gMOP of the conflict is added to the gMOPs of all the other events. In between (e.g. in the Dempster-Shafer case) both the ignorance and the other events get a proportional part of the uncertainty caused by the conflict.

Example 10.5

There are two sensors, both give mass of probability values for 3 events (A , B and C), and for the ignorance ($\{A, B, C\}$). The values provided by the sensors in a given time are shown in Table 10. . Let us see how the combined MOP depends on the k parameter of the Inakagi's unified combination rule.

Event (•)	A	B	C	0, conflict	Θ=Ignorance={A, B, C}
$q^{(1)}(\bullet)$	0.2	0.1	0.2		0.5
$q^{(2)}(\bullet)$	0.3	0.2	0.1		0.4
$q^{(1\oplus 2)}(\bullet)$	0.29	0.16	0.15	0.2	$0.5 \cdot 0.4 = 0.2$
$m_U^{(1\oplus 2)}(\bullet)$	$[1 + k \cdot 0.2] \cdot q^{(1\oplus 2)}(\bullet)$				$[1+k \cdot 0.2] \cdot 0.2 + [1 + k \cdot 0.2 - k] \cdot 0.2 = 0.4 - 0.12 \cdot k$

$$0 \le k \le \frac{1}{1 - q^{(1\oplus 2)}(0) - q^{(1\oplus 2)}(\Theta)} = \frac{1}{1 - 0.2 - 0.2} = 1.67$$



45. ábra. Illustration to the Example 10.5.

11.3. 10.3 Applications of data fusion

There are several areas where sensor data is used.

One of the most important ones is the target tracking of aircrafts, ships etc.

Another interesting field is the human-computer interaction, where context-aware computing tries to make computers to understand the physical environment, including people around.

In recent years wireless sensor technology has opened the possibility to connect large number of sensors in one or more networks. In this framework the sensor data fusion is a natural requirement.

12. 11 User's behavioral modeling

Intelligent embedded systems (IES) should be intelligent in their interactions with other intelligent systems, including us, humans. In turn, an IES should develop a behavioral model of its human partners and it should interact, or communicate according to the model and the (common) goals and subject to the methods it may apply. The list of these methods is extremely broad. One end is an intelligent writing system, such as the predictive text-on-9-digits, or T9 system[1], or Dasher[2]. The other end is a neural prosthetic device, including deep brain stimulation[3] to counteract, e.g., Parkinson's disease, or a cortical implant [4] for direct mind-machine interaction. Such interfaces will be considered in the next Chapter.

In this Chapter we start by considering human communication since it is the very evolutionary design that transmits and receives behavioral information of one human to the other. Then we consider methods for measuring and registering latent behavioral signals and finally we establish concepts for modeling behavioral dynamics and the optimization of interaction.

12.1. 11.1 Human communication

There has been and advantage of forming groups during evolution. An example is the group of birds when flying; it consumes less energy. Beyond such relatively simple advantages, other advantages have emerged due to the multiplication of sensory information and the increased strength of orchestrated actions. Such information sharing and (distributed) decision making is impossible without communication. Communication - alike to engineering - requires a protocol. The name protocol originates from human culture and in fact, communication

[1]http://en.wikipedia.org/wiki/T9_(predictive_text)
[2]http://www.inference.phy.cam.ac.uk/dasher/
[3]http://en.wikipedia.org/wiki/Deep_brain_stimulation
[4]http://en.wikipedia.org/wiki/Brain-computer_interface

can be seen as (part of the) culture. A recent book [] puts culture into an evolutionary perspective and claims that human nature and human societies can be understood from primate social evolution

There are many ways how we communicate. The design is evolutionary so it is arguably optimal in two ways: this is the best system evolution could develop for making us competitive and this is the communication system that we learn and use from early childhood. The scientific discipline, anthroposemiotics[5] studies this field. According to the nomenclature, there are at least 5 types of communications, out of which we are interested in the following three:

- intra-personal communication: the user communicates with her/himselves. This type belongs to cognition, it could be related to emotional, explanatory, or planning activities among many others and their combinations. The movie, Fiddler on the Roof is full with examples.
- interpersonal communication
- communication in a groups that aims to influence group dynamics

These types are barely separable; they are all related to latent/hidden intentions and emotions and can be decoded in terms of such intentions and emotions. This is called mind reading or the ability to develop a 'theory of mind', i.e., the ability to attribute mental states about beliefs, intents, desires, pretending, knowledge, and so on to others and to ourselves. This is also a philosophical issue since the mind as such is not measurable and it is available for the self only (cf. philosophy of mind[6]). For the sake of gaining insights from exaggeration, we say that autism 'can be understood' on the basis that autistic people are not interested in other people's mind, or may try to save their mind intact from the influence of other people, whereas one might claim that the schizophrenic mind is multi-valued.

Mind reading is so important that infants have special feedforward perception-action system at birth that alleviates the association of other people's feelings to the own feelings and to the respective actions. Infants and imitation have been studied for a long time. The practice of newborns that they can imitate mouth opening and tongue protruding without seeing what they are doing can be interpreted in many ways ([]), but learning. Some may claim that laugh, on the other hand, has some cross modal acoustic feedback that helps the associative mapping, but the value of the acoustic feedback can be debated. Thus, early imitations are most probably inherited. Since they could be learned easily at some later time, these early imitations should be most relevant for the parent, especially for the mother to detect the problems of the infant. One can guess that infants whose mother is not good in mind reading have lower survival rates and females might be better in mind reading than males. In fact, autistic people are also called 'extreme males' and the female portion of the autistic population is very low.

12.2. 11.2 Behavioral signs

12.2.1. 11.2.1 Paralanguage and prosody

Emotional signs can modulate pitch, volume and the intonation of speech. The tone of voice may suggest anger, surprise, or happiness. Gasp, a sudden and sharp inhalation of air through the mouth may indicate surprise, shock, or disgust and although it can be produced intentionally, it is rarely intentional. A gasp is typically followed by sigh that releases the air inhaled during gasp. Such signs of emotions form part of paralanguage. Prosody can be divided into intonation units (also called prosodic units); a segment of speech that occurs during a single pitch and rhythm contour.

Paralinguistic cues include loudness, speed of talk, pitch with the contour of pitch. They give information about the emotions or attentions. They can be either intentional or unintentional and people can hide or fake emotional such signals.

12.2.2. 11.2.2 Facial signs of emotions

[5]http://en.wikipedia.org/wiki/Human_communication
[6]http://en.wikipedia.org/wiki/Philosophy_of_mind

Facial signs of emotions are very expressive. It is intriguing that it is easier to recognize the facial expression of a person who is approaching from a distance than the identity of that person. Also, facial expressions especially their temporal changes can represent a number of mental states. Some of the facial expressions, namely the basic emotional states, like anger, disgust, sadness, fear, contempt, happiness, surprise, are considered culture independent and general for the human race. In the case of anger the eyebrows are pulled down, upper lids are pulled up, lower lids are pulled up, margins of lips are roller in, and lips may be tightened. Happiness make the muscles around the eyes tightened, gives rise to wrinkles around the eyes, makes the cheeks raised and raises the lip corners diagonally. Similar descriptions can be given for the other five basic emotions.

The original story goes back to the sixties of the last century, when psychologists Paul Ekman and Wallace Friesen studied Fore, an isolated, preliterate culture in New Guinea. They told stories to a group of Fore describing different basic emotions, such as happiness, sadness, fear, and disgust. Then they asked the Fore to match emotional pictures to the stories. It turned out that Fore classified facial expressions alike to other people except that they could not distinguish fear and surprise, see [] and the references therein.

These facial changes are produced by the muscles of the face giving rise to textural changes (like producing wrinkles) and changes in the shape of the face. The number of muscles is close to hundred (Fig. 46)[7]. A broad variety of facial expression can be produced and they may gain different meanings in different cultures. Furthermore, although the basic emotions are very similar and can be recognized everywhere, there are subtle differences that come with culture. People can distinguish, e.g., Japanese-Americans (American people of Japanese heritage) and Japanese nationals by their smiles [és]. On the other hand, tourism and movies may make an impact on facial expressions and cross cultural differences may change by time.

[7]http://en.wikipedia.org/wiki/Facial_Action_Coding_System

AU Number	FACS Name	Muscular Basis
0	Neutral Face	
1	Inner Brow Raiser	frontalis (pars medialis)
2	Outer Brow Raiser	frontalis (pars lateralis)
4	Brow Lowerer	depressor glabellae, depressor supercilii, corrugator supercilii
5	Upper Lid Raiser	levator palpebrae superioris
6	Cheek Raiser	orbicularis oculi (pars orbitalis)
7	Lid Tightener	orbicularis oculi (pars palpebralis)
8	Lips Toward Each Other	orbicularis oris
9	Nose Wrinkler	levator labii superioris alaeque nasi
10	Upper Lip Raiser	levator labii superioris, caput infraorbitalis
11	Nasolabial Deepener	zygomaticus minor
12	Lip Corner Puller	zygomaticus major
13	Sharp Lip Puller	levator anguli oris (also known as caninus)
14	Dimpler	buccinator
15	Lip Corner Depressor	depressor anguli oris (also known as triangularis)
16	Lower Lip Depressor	depressor labii inferioris
17	Chin Raiser	mentalis
18	Lip Pucker	incisivii labii superioris and incisivii labii inferioris
19	Tongue Show	
20	Lip Stretcher	risorius w/ platysma
21	Neck Tightener	platysma
22	Lip Funneler	orbicularis oris
23	Lip Tightener	orbicularis oris
24	Lip Pressor	orbicularis oris
25	Lips Part	depressor labii inferioris, or relaxation of mentalis or orbicularis oris
26	Jaw Drop	masseter; relaxed temporalis and internal pterygoid
27	Mouth Stretch	pterygoids, digastric
28	Lip Suck	orbicularis oris
29	Jaw Thrust	
30	Jaw Sideways	
31	Jaw Clencher	masseter
32	[Lip] Bite	
33	[Cheek] Blow	
34	[Cheek] Puff	
35	[Cheek] Suck	
36	[Tongue] Bulge	
37	Lip Wipe	
38	Nostril Dilator	nasalis (pars alaris)
39	Nostril Compressor	nasalis (pars transversa) and depressor septi nasi
41	Glabella Lowerer	Separate Strand of AU 4: depressor glabellae (aka procerus)
42	Inner Eyebrow Lowerer	Separate Strand of AU 4: depressor supercilii
43	Eyes Closed	Relaxation of levator palpebrae superioris
44	Eyebrow Gatherer	Separate Strand of AU 4: corrugator supercilli
45	Blink	Relaxation of levator palpebrae superioris; contraction of orbicularis oculi (pars palpebralis)
46	Wink	orbicularis oculi

46. ábra. Facial Action Coding System

12.2.3. 11.2.3 Head motion and body talk

Head plus eye motion can be very expressive for emotions and intentions. One may 'point' with the eyes, the head or both giving instructions, for example. Also, behavioral signs, including head motion characterize the individual. Hill and Johnston [] showed that the recognition of identity is helped more by rigid motion cues (such as head motion) than by non-rigid motion cues (such as facial expression). Recognition of sex, however, seems to be mediated by changes in facial expressions. Head motion codes are shown in Fig. 47[8].

[8]http://en.wikipedia.org/wiki/Facial_Action_Coding_System

AU Number	FACS Name	Action
51	Head Turn Left	
52	Head Turn Right	
53	Head Up	
54	Head Down	
55	Head Tilt Left	
M55	Head Tilt Left	The onset of the symmetrical 14 is immediately preceded or accompanied by a head tilt to the left.
56	Head Tilt Right	
M56	Head Tilt Right	The onset of the symmetrical 14 is immediately preceded or accompanied by a head tilt to the right.
57	Head Forward	
M57	Head Thrust Forward	The onset of 17+24 is immediately preceded, accompanied, or followed by a head thrust forward.
58	Head Back	
M59	Head Shake Up and Down	The onset of 17+24 is immediately preceded, accompanied, or followed by an up-down head shake (nod).
M60	Head Shake Side to Side	The onset of 17+24 is immediately preceded, accompanied, or followed by a side to side head shake.
M83	Head Upward and to the Side	The onset of the symmetrical 14 is immediately preceded or accompanied by a movement of the head, upward and turned and/or tilted to either the left or right.

47. ábra. Head Motion Coding System

12.2.4. 11.2.4 Conscious and subconscious signs of emotions

Situations (spatio-temporal context) can help the recognition of emotions and mental states, the hidden variables of behavior. It is easy to mix for example anger and being suspicious during problem solving, but these expressions can be set apart in temporal context and according to the success or failure. Many facial expressions and head/eye motions are unconscious and are hard (sometimes impossible) to hide. Polygraph, that measures physiological indices including blood pressure, respiration and skin conductivity, is a commonly used tool for lie detection. It is considered unreliable by many people []. Ekman claims to reach 90% detection accuracy when facial expressions are combined with voice and speech measures. This claim needs to be verified by others, but the key message is that deception can not be identified by a single clue Electroencephalograph (EEG) and functional Magnetic Resonance Imaging (fMRI) are also capable to infer about deceptive behavior. We have limited control over many of these signals.

The detection of pain or tiredness is a more important issue in the context of of human-computer interactions and collaboration if we assume that the computer works for the sake of the user and that the user would like to take full advantage of the a backing statistics based recommender system. It is equally relevant to identify situations when the user is "in the zone" or in the "state of flow" []; a completely focused motivation. This is a single minded immersion when emotions serve performance and learning. Other terms that try to capture this state include in the moment, on a roll, wired in, in the groove, on fire, in tune, centered, or singularly focused [9]. The facial expression, however, is very similar a blank or oblivious look. So while flow is to be achieved in the educational setting, the unmindful state is to be avoided, but the facial expressions are very similar. A distinction can be by means of the spatio-temporal context and by previous experiences on user behavior in the context of the actual task.

In sum, human-computer collaboration requires modeling of the behavior including as much information as possible starting from visual and acoustic information and taking advantage of other sensors, like blood pressure, skin conductance, heart rate. The availability of such additional signal has been increased drastically by the fast evolution of mobile tools and mobile phones.

12.3. 11.3 Measuring behavioral signals

We will review acoustic and visual behavioral signals since they are easily available and they are the typical forms of human-human communication. The best techniques of our days utilize databases and tune the recognition / identification / classification by means of large databases. There is a related European project, the

[9]http://en.wikipedia.org/wiki/Flow_(psychology)

'European network of excellence in social signal processing'[10] that also lists some of the most relevant databases[11] and a number of tools[12] for such studies. Some homeworks will be selected from these.

12.3.1. 11.3.1 Detection emotions in speech

This topic is in the focus of current interest to improve user experiences with automated phone attendants. A large set of methods have been tried and demonstrated success. For a review, see [és] and the references therein. Special challenges have been organized to compare the methods and the databases [].

12.3.2. 11.3.2 Measuring emotions from faces through action units

Below, we review the most popular models of facial tracking. Then we turn to the estimation methods.

12.3.2.1. 11.3.2.1 Constrained Local Models

CLM methods are generative parametric models for person-independent face alignment. In this work we were using a 3D CLM method, where the shape model is defined by a 3D mesh and in particular the 3D vertex locations of the mesh, called landmark points. Consider the shape of a 3D CLM as the coordinates of 3D vertices of the M landmark points:

$$\mathbf{x} = (x_1, y_1, z_1, ..., x_M, y_M, z_M)^T, \qquad (1)$$

or, $\mathbf{x} = (\mathbf{x}_1^T, \ldots, \mathbf{x}_M^T)^T$, where $\mathbf{x}_i = (x_i, y_i, z_i)^T$. We have N samples: $\{\mathbf{x}^{(n)}\}_{n=1}^N$. CLM models assume that - apart from the global transformations; scale, rotation, and translation - all $\{\mathbf{x}^{(n)}\}_{n=1}^N$ can be approximated by means of the linear principal component analysis (PCA) forming the PCA subspace. Details of the PCA algorithm are well covered by Wikipedia https://en.wikipedia.org/wiki/Principal_component_analysis. The interested reader may wish to investigate the more elaborated tutorial []

In the next subsection we briefly describe the 3D Point Distribution Model and the way CLM estimates the positions of the landmarks.

12.3.2.2. 11.3.2.2 Point Distribution Model

The 3D point distribution model (PDM) describes non-rigid shape variations linearly and composes it with a global rigid transformation, placing the shape in the image frame:

$$\mathbf{x}_i(\mathbf{p}) = s\mathbf{P}\mathbf{R}(\bar{\mathbf{x}}_i + \mathbf{\Phi}_i\mathbf{q}) + \mathbf{t}, \qquad (2)$$

where $i = 1, \ldots, M$, $\mathbf{x}_i(\mathbf{p})$ denotes the 2D location of the i^{th} landmark subject to transformation $\mathbf{P}$, and $\mathbf{p} = \{s, \alpha, \beta, \gamma, \mathbf{q}, \mathbf{t}\}$ denotes the parameters of the model, which consist of a global scaling s, angles of rotation in three dimensions ($\mathbf{R} = \mathbf{R}_1(\alpha)\mathbf{R}_2(\beta)\mathbf{R}_3(\gamma)$), translation $\mathbf{t}$ and non-rigid transformation $\mathbf{q}$. Here $\bar{\mathbf{x}}_i$ is the mean location of the i^{th} landmark averaged over the database, i.e. $\bar{\mathbf{x}} = [\bar{\mathbf{x}}_1^T; \ldots; \bar{\mathbf{x}}_M^T]$, $\bar{\mathbf{x}}_i = [\bar{x}_i, \bar{y}_i, \bar{z}_i]^T$, $\bar{x}_i = \frac{1}{N}\sum_{n=1}^N x_i^{(n)}$, and similarly, for $\bar{y}_i$ and $\bar{z}_i$. Matrix $\mathbf{\Phi}_i$ $(i = 1, \ldots, M)$ is a $3 \times d$ piece in $\mathbf{\Phi} \in \mathbb{R}^{3M \times d}$ and corresponds to the landmarks. Columns of $\mathbf{\Phi}$ form the orthogonal projection matrix of principal component analysis and its compression dimension is d. Finally, matrix $\mathbf{P}$ denotes the projection matrix to 2D:

[10]http://sspnet.eu/
[11]http://sspnet.eu/category/sspnet_resource_categories/resource_type_classes/dataset/
[12]http://sspnet.eu/category/sspnet_resource_categories/resource_type_classes/tool/

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \qquad (3)$$

and thus $\mathbf{x}_i(\mathbf{p}) \in \mathbb{R}^2$ ($\forall i$).

By applying PCA on the $\mathbf{x}_i$ points we get an estimate of the prior of the parameters:

$$p(\mathbf{p}) \propto N(\mathbf{q}; \mathbf{0}, \mathbf{\Lambda}), \qquad (6)$$

that is CLM assumes a normal distribution with $\mathbf{0}$ mean and $\mathbf{\Lambda}$ variance for parameters $\mathbf{q}$. $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1; \ldots; \lambda_d) \in \mathbb{R}^{d \times d}$ in (6) is provided by the PCA and the parameter vector assumes the form $\mathbf{p} = [s; \alpha; \beta; \gamma; \mathbf{t}; \mathbf{q}]$.

12.3.2.3. 11.3.2.3 Formalization of Constrained Local Models

CLM is constrained through the PCA of PDM. It works with local experts, whose opinion is considered independent and are multiplied to each other:

$$J(\mathbf{p}) = p(\mathbf{p}) \prod_{i=1}^{M} p(l_i = 1 | \mathbf{x}_i(\mathbf{p}), \mathcal{I}), \qquad (7)$$

where $l_i \in \{-1, 1\}$ is a stochastic variable, which is 1 (-1) if the i^{th} marker is (not) in its position, $p(l_i = 1 | \mathbf{x}_i(\mathbf{p}), \mathcal{I})$ is the probability that for image $\mathcal{I}$ and for marker position $\mathbf{x}_i$ determined by parameter $\mathbf{P}$, the i^{th} marker is in its position.

Local experts are built on Logit Regression and are trained on labeled samples. The functional form of Logit is

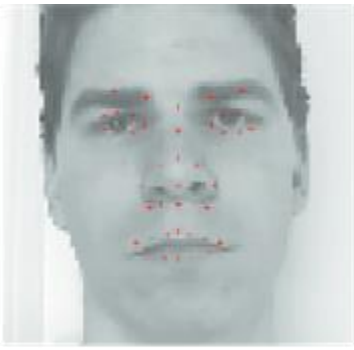
$$p(l_i = 1 | \mathbf{y}_i, \mathcal{I}) = \frac{1}{1 + e^{\mathbf{w}_i^T N(\mathcal{I}(\mathbf{y}_i)) + b_i}}, \qquad (8)$$

where $N(\mathcal{I}(\mathbf{y}_i))$ is a normalized image patch around point $\mathbf{y}_i$, $\mathbf{w}_i$ and b_i are parameters of the distribution to be learned from samples. Positive and negative samples for the right corner of the right eye are shown in Fig. 48.

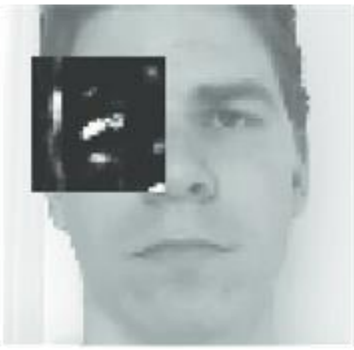
Local expert's response - that depend on the constraints of the PDM and the response map of the local expert in an appropriate neighborhood - can be used to express $J(\mathbf{p})$ in (7) (Fig. 49):

$$J(\mathbf{p}) = p(\mathbf{p}) \prod_{i=1}^{M} \sum_{\mathbf{y}_i \in \Psi_i} p(l_i = 1 | \mathbf{y}_i, \mathcal{I}) \, p(\mathbf{y}_i | \mathbf{x}_i) \, \mathcal{N}(\mathbf{y}_i; \mathbf{x}_i, \rho \mathbf{I}) \qquad (9)$$

where CLM assumes $\mathbf{y}_i = \mathbf{x}_i + \epsilon_i$ with $\epsilon_i \sim \mathcal{N}(\mathbf{0}, \rho \mathbf{I})$, $\rho = \sum_{i=d+1}^{3M} \frac{\lambda_i}{(3M-d)}$, λ_i is the i^{th} eigenvalue of $Cov(\mathbf{x})$, the covariance matrix of stochastic variable $\mathbf{x}$ and where we applied Bayes'rule and the tacit assumption [] that $\sum_{\mathbf{y}_j \in \Psi_i} \mathcal{N}(\mathbf{y}_i; \mathbf{x}_i, \rho \mathbf{I})$ is a weak function of the parameters to be optimized was accepted.



48. ábra. (a): Landmarks on the face. (b): response map for the right corner of the right eye.



49. ábra. (a): Landmarks on the face. (b): response map for the right corner of the right eye.

12.3.2.4. 11.3.2.4 Active Appearance Models

Both two- and three-dimensional Active Appearance Models (AAMs) have been developed. They are made of an active shape model, which is similar to the probability distribution model of the CLM, and a texture model, which is radically different. In this latter, one takes the marker points, connects those by lines in such a way that markers form the vertices of triangles and all closed areas are triangles. The texture within the triangles undergo affine transforms in the matching procedure to match actual estimations of the triangles. Both the texture model and the shape model are compressed and Gaussian distribution is assumed for the joined model []

12.3.2.5. 11.3.2.5 Estimation of action units

Estimation of action units (AUs) utilize annotated databases; the images are annotated by the AUs and by their strength. If the CLM fit is satisfactory then one may estimate the AUs from the (change of the) shape, or from the (change of the) texture around the marker points. Combined methods have been tried in the literature and show some improvements.

12.3.2.6. 11.3.2.6 Emotion estimation

Similar estimation can be used for the emotions. One may use directly the emotion labeled faces together with the CLM fit of the marker points to estimate shape changes and/or changes od the texture around the marker points. Typical estimations make use of SVM based linear regressors and SVM classifiers both for AUs and for emotions.

12.3.3. 11.3.3 Measuring gestures

Measuring gestures and body motions are possible in a number of ways, such as (i) via wearable digital textile sensor [] that may also include accelerometer and gyroscope [] as well as other sensors, e.g. ECG that may save life[13]. Kinect is a useful tool for remote optical measurement. Gesture estimation from a single camera video capture is demanding but it is the only available tool for the annotation of movies and videos. While the case of Kinect is relatively simple since it gives back a 3D view, single camera systems may take advantage of structure from motion algorithms. Two-camera stereo vision or its many camera generalization can use registration methods, e.g., using the silhouettes of the arms or the body if those are not occluded.

12.4. 11.4 Architecture for behavioral modeling and the optimization of a human computer interface

At a very high level, the intelligent architecture that aims to model and possibly to optimize human performance is made of the following components: (i) sensory processing unit, (ii) control unit, (iii) inverse dynamics unit, and (iv) decision making unit. Although it looks simple, one has to worry about a number of things, such as the continuity of space and time, the curse of dimensionality, if and how space and time should be discretized, and planning in case of uncertainties, e.g., in partially observed situations, including information about purposes, cognitive, and emotional capabilities of the user. Below, we review the basic components of the architecture. Every component can be generalized to a great extent. In particular cases, some of the components may be left out. The architecture should be able to estimate parameters of human behavior.

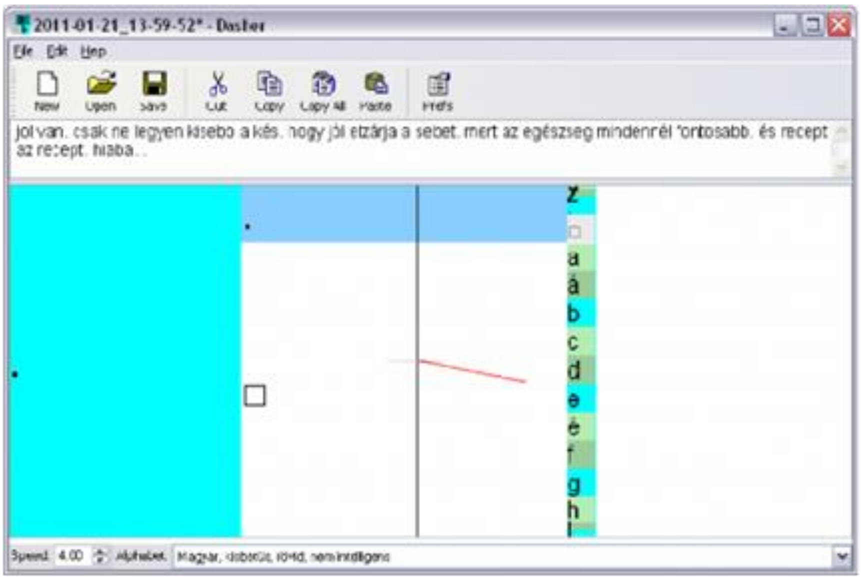
- This stage selects different samples under random control in order to collect state-action-new state triples in order to learn the controllable part of the space.
- For a sufficient number of collected samples, in principle, one can estimate the dimension of the state space and the related non-linear mapping of sensory information to this lower dimensional manifold. In the present example, the low dimensional manifold is known and selected samples will be embedded into the low dimensional space will be used for interpolation.
- Out-of-sample estimations will be used for the identification of the dynamics in the form of autoregressive exogeneous (ARX) process. Generalization to more complex non-linear models, such as switching non-linear ARX models is, in principle, possible.
- The ARX process can be inverted and the inverted ARX process can be used for control.
- The inverse dynamics can be learned. A linear-quadratic regulator[14] is a relatively simple option.
- Optimization concerns long-term goals or a hierarchy of those. Optimization then belongs to the field of reinforcement learning (RL). The continuity of space and actions can be overcome by means of the event learning formalism [és] of RL that enables continuous control in the optimization procedure.
- RL optimization can be accomplished in many ways, including the Optimistic Initial Model, which is favorable for many variable cases [].

12.4.1. 11.4.1 Example: Intelligent interface for typing

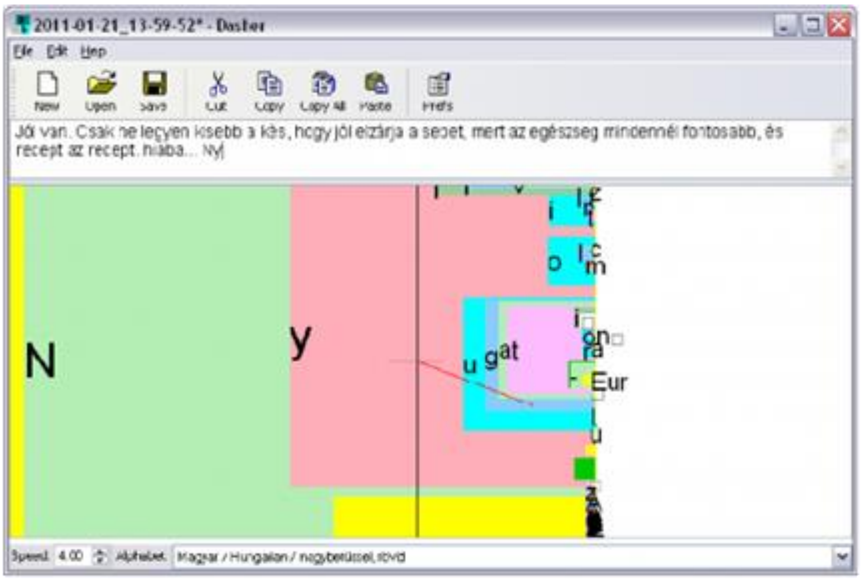
This is an introductory example with the aim to include the components of the behavioral modeling and optimization components described before. The example is a prototype for more complex modeling and interfaces. This example is about the modeling and the optimization of the human control in a particular task. It could also take advantage of facial expressions, eye movements and alike.

The example concerns performance optimization when using the Dasher writing tool[15]. Dasher has been designed for gaze control and can be used efficiently head pose control. Dasher interface is shown in Figs. 50 and 51. [].

[13]http://en.wikipedia.org/wiki/Smart_shirt
[14]http://en.wikipedia.org/wiki/Linear-quadratic_regulator
[15]http://www.inference.phy.cam.ac.uk/dasher/



50. ábra. Dasher without database and prediction



51. ábra. Dasher with prediction by partial matching [?]

Dasher can be characterized roughly as a zooming interface. The user zooms in at the point where s/he is pointing to by using the cursor. The image, which is subject of zooming is made of letters, so that any point you zoom in corresponds to a piece of text. Zooming is complemented by moving the text opposite to the cursor. The more one zooms in on the right hand side of the image, the longer the piece of text that crosses the line of the cursor and gets written. Corrections can be made by moving the cursor to the left hand side of the image.

The interface is made efficient by a predictive language model. This language model determines the size of the area a letter has. Probable pieces of text are given more space, so they are quick and easy to select. Improbable pieces of text are given less space, so they are harder to write. According to experiments learning to use the writing tool takes time and gives rise to certain practices that may change from user to user []. The goal of optimization is to adjust the cursor position in such a way that writing speed is optimized for average writing speed. This requires the estimation of head pose with its changes as well as the optimal adjustment of the cursor.

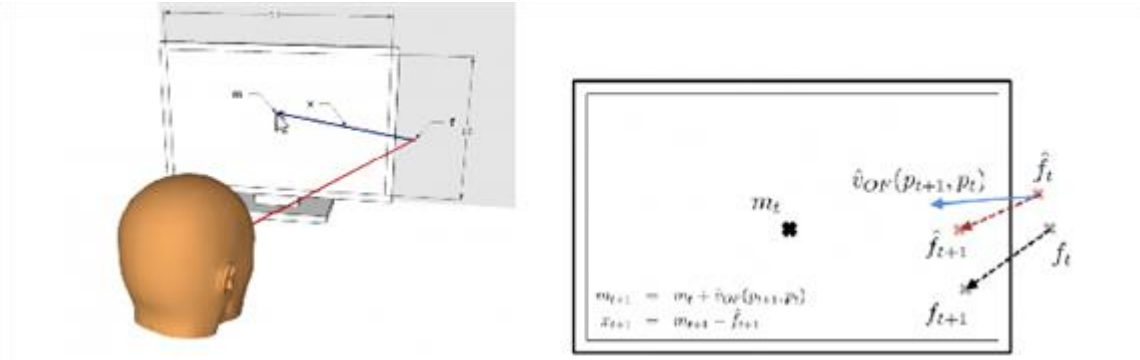
Pose estimation can take advantage of Principal Component Algorithm for shape, texture, and details. For more precise pose estimation the CLM or AAM tools of the previous section can be utilized. The first step is the localization of the face by means of the so called Viola-Jones face detector. Relative changes of the pose can take advantage of optic flow estimation, respectively. Given the pose estimation, the input to the learning algorithm can be made by hand in the present case: denote the screen size normalized position of the cursor by $(m_x, m_y) \in [0,1]^2$ and the estimation of the two dimensional position of the head by $(f_x, f_y) \in \mathbb{R}^2$. The two-dimensional vector $x = [m_x - f_x, m_y - f_y]^T \in \mathbb{R}^2$ can be taken as an estimation of the state for the present control task.

12.4.2. 11.4.2 ARX estimation and inverse dynamics in the example

The AR model assumes the following form

$$\begin{aligned} x_{t+1} &= m_{t+1} - f_{t+1} && (10) \\ m_{t+1} &= m_t + v_t && (11) \end{aligned}$$

where $m_t \in \mathbb{R}^2$ is the position of the cursor at time t, $f_t \in \mathbb{R}^2$ is the point where the roll axis of the pose hits the screen as shown in Fig. 52, $v_t \in \mathbb{R}^2$ is the speed vector of the projected f_t on the screen over unit time and no additional noise was explicitly assumed. We have direct access to the cursor position and need to estimate the other parameters. Since $v_t = f_{t+1} - f_t$ it follows that $x_{t+1} = x_t$ in the absence of estimation errors and control. The goal is to control and optimize x_t for writing speed.



52. ábra. Illustration and parameters of the experiments. Left: experimental arrangement. m: cursor position, f: position where the roll axis of head pose crosses the screen. x: 'observation'. Right: True and estimated quantities at time t. $v_{OF}(p_{t+1}, p_t)$: optic flow based estimation of the motion vector f. p_t: positions of feature points on the 2D projected face at time t

We do not have direct access to f_t or x_t, but use their estimations $\widehat{f}_t \in \mathbb{R}^2$ and $\widehat{x}_t$ through the measurement of the optic flow (Fig. 52) of the face on subsequent image patches $(\widehat{v}_{OF}(p_{t+1}, p_t))$, $p_t \in \mathbb{R}^{2k}$ denotes the 2D coordinates of k characteristic points (Fig. 53) within the facial region of the image

53. ábra. Tools for human-computer interaction: features for optic flow estimation (green markers), eye tracker and head pose estimation (yellow marker).Left: full face. Right: close-up

Collecting a number of data $\widehat{x}_1, \ldots, \widehat{x}_T$, one can estimate the unknown parameters of matrix B by direct control, using distances on the screen as $\widehat{x}_{t+1} = \widehat{x}_t + Bu_t + n_t$ and then inverting it to yield desired state x_d: $u_t = \widehat{B}^{-1}(\widehat{x}_d - \widehat{x}_t)$. Inserting the result back to the ARX estimation one has $\widehat{x}_{t+1} \approx \widehat{x}_d$. Note that this inverse dynamics can be extended to sophisticated non-linear 'plants' if needed.

12.4.3. 11.4.3 Event learning in the example

Now, we define the optimization problem. For this, we transcribe the task into the so called event learning framework that works with discrete states, provides the actual state and desired successor state to a backing controller. Then the controller tries to satisfy the 'desires' by means of the inverse dynamics. For a given experienced state i and its desired successor state i^+, where $i, i^+ = 1, 2, \ldots, N$ and N is the number of states, that is, for a desired event $e(i, i^+)$, the controller provides a control value or a control series. The estimated value E^{π}_{i,i^+} of event $e(i, i^+)$ denotes the estimated long-term cumulated discounted reward under a fixed policy, i.e., a mapping $\pi = \pi(i, i^+)$, Then, the event learning algorithm learns the limitations of the backing controller and can optimize the policy in the event space [].

12.4.4. 11.4.4 Optimization in the example

Many optimization methods are available for the optimization of events in the sake of the maximization of long-term cumulated reward. One option is the so called optimal initial model (OIM) []. OIM aims at resolving the exploration exploitation dilemma; i.e., the problem if new events are to be sought for or if the available knowledge should be exploited for the optimization without further exploration.

The example of this section concerned the optimal personalization of a human-computer interface that learns the specific features of user behavior and adapts the interface accordingly. Related homeworks and thesis works are put forth in the next section.

12.5. 11.5 Suggested homeworks and projects

1. Action Unit Studies:

   a. AU detector: download the AU detector called LAUD[16]. A set of movies about facial expressions will be made available for this homework. Task: using the detected AUs determine if a basic emotions is present or not.

   b. Vowel detector: use LAUD to identify vowels from speech. Use the VLOG(s) provided during the course and use a simple classifier, e.g., a linear Support Vector Machine on the AU data.

[16] http://ibug.doc.ic.ac.uk/resources/laud-programme-20102011/

c. critics to LAUD: determine the limitations of LAUD (angle, light conditions, occlusions)

d. Improve LAUD: use spatio-temporal tools including Hidden Markov Models and Sparse Models to improve recognition accuracy.

2. Algorithm and sensor comparison:

a. AAM and CLM: compare the AAM-FPT, i.e., the Active Appearance Model based Facial Point Tracker[17] of SSPNETwith the MultiSense software based on Constrained Local Model[18]

b. 3D CLM and Kinect based CLM: compare the performance of the CLM if the input is from a single webcam or from a Kinect device. Explain the differences.

3. Gesture recognition:

a. Gesture recognition: select three arm gestures from SSPNET. Use the Kinect SDK and collect data. Build a recognition system using Radial Basis Functions to recognize the three gestures.

b. Rehabilitation: take a look to the 'Fifth Element Project'[19]. Design a scenario that helps to loosen the shoulders. Take advantage of internet materials, like http://www.livestrong.com/article/84763-rehab-shoulder-injury/

4. Suggested thesis works in the area of modeling and optimization of human-computer interaction. Discuss them with your supervisor:

a. Dasher: Redo the Dasher project []. The optimization can be improved. Make suggestions, select and design with your supervisor, execute the project, take data and analyze them.

b. Head and eye motion: Take video of you own face during working (same environment, same chair, different times). Label the activities (thinking, tired, focusing, reading, working, 'in zone', etc.) Build classifiers and try to identify the signs of the different behavioral patterns. Develop a description that fits your behavior better. Compute the information you gain from the different components for classification.

c. Use computer game Tetris. Recruit 10 people for the study. Measure their activity patterns and compute the correlations with the few important events of Tetris (hard situation, making a mistake, deleting a row, deleting many rows). Cluster the users.

d. Optimize Tetris for the user. The task is the same as above with the 'slight' difference that you want to keep the user 'in the zone', i.e., in the state when s/he is focusing the most. Your control tool is the speed of the game.

e. Optimize Tetris for facial expressions. The more facial expressions you detect, the better your program. Your tool is the probability of the different blocks. Your action is that you can change these probabilities during the game. Make a list of possible user behaviors before starting the experiments and limit the exploration exploitation to these user models.

13. 12 Questions on human-machine interfaces

In the previous section (Chapter 11) a number of issues concerning human-machine interfaces have been mentioned. Now, we shall deal with the related questions concerning safety, privacy, data-sharing, recommender systems. Most of these questions involve ethical and legal issues, and may have impact on our health, well-being, personal life, etc. These are complex problems that need to be treated. Here, we are limited by both space and knowledge; many of these questions are open and are subject to hot debates.

13.1. 12.1 Human computer confluence: HC_2

[17] http://sspnet.eu/2011/03/aam-fpt-facial-point-tracker/
[18] https://vhtoolkit.ict.usc.edu/plugins/viewsource/viewpagesrc.action?pageId=13566443
[19] http://fifthelementproject.com/

Human computer confluence all kinds of human-computer interactions, including visible, invisible, explicit, implicit, embodied, and implanted interactions between humans and system components. New classes of user interfaces are evolving that make use of several sensors and are able to adapt their physical properties to the current situational context of users.

Human Computer Confluence is to become a research priority in "Horizon 2020" (2013-2020), the funding programme of the European Commission that follows the 7th Framework Programme (FP7, 2007-2013). Key research challenges of HC2 include the extension of human perception, e.g., by augmented reality compact lenses and infrared sensitive retinas, and the development of cognitive prostheses, an early version being the motor cortex microelectrode array for motion disabled people. Other areas cover, include, and influence empathy and emotion, well-being and quality of life, socially inspired technical systems and value sensitive design.

The information that is to be analyzed is huge and a large part of the data will be subject to 'single pass' analysis since it can't be kept and should be dropped immediately. Selected and compressed portions of the data will form what we call today as BIG DATA,large and linked datasets including those obtained by data harvesting across heterogeneous data sources. There is a strong need for collaboration between social science scholars, open data activists, statisticians, computer scientists and other relevant parties in order to design a data environment capable of amplifying positive externalities and reducing negative externalities. Positive externalities to be addressed include (but are not limited to) economic and legal models for efficient data markets and negative externalities include (but are not limited to) the privacy risks that come from the reidentification of personal information, particularly as a consequence of more and more data sets becoming available and being linked to one another. Ethical and moral considerations should also be taken into account.

In the enxt section brain reading tools are sketched with the note that the field is developing very quickly and the tools become obsolete in about three years or so.

13.2. 12.2 Brain reading tools

Brain reading tools convey information about the internal processes of the brain and the state of the brain. Some of them are simple, and human interpretable, like emotion monitoring optical devices. Others are more complex and are harder to interpret, such as devices recording the electrical activity along the scalp. This is the field of electroencephalography. Signals from deeper regions are brought together by magnetography that can monitor regions still close to the scalp or - with high magnetic fileds - can monitor the whole brain. These are passive tools. Microelectrode arrays can be built into the brain and can serve both for monitoring and for influencing neurons of different crtical areas. Some of these options - without the aim of completeness - are detailed (to a limited extent) below.

The interested reader is referred to the main sites of these subjects on the Internet; the field is developing so quickly that this is the best route to find the most reliable and still valid information. We mention the following sites: First of all, look up Wikipedia, which is being edited by lots of experts and Scholarpedia, which is a bit delayed, but it is peer reviewed and the content is more reliable but the coverage is smaller. Beyond these, there are special sites, such as Kurzweil's site on 'accelerating intelligence' and the site of the Lifeboat Foundation that aims to safeguard humanity. Another information source is TED; TED invites the best experts who explain their thoughts and discoveries and these talks are clear, up-to-date and understandable for the non-experts.

13.2.1. 12.2.1 EEG

Electro-encephalography is relevant since (9) it can bring signals from the surface of the brain, so mostly from the grey matter and (ii) due to the appearance of dry electrodes, it has become available in the form of jewel-like gadgets (Fig. 54)

54. ábra. EEG headsets. Left: tools for scientific research. Right: everyday tools and gadgets

The EEG tools have been used for controlling an exoskeleton, see the report on Mind-controlled exoskeleton that helps people to walk again or the gadgets sold by NeurSky that can be used for game playing, or other gaming tools like the the Force Trainer.

13.2.2. 12.2.2 Neural prosthetics

Neurospace a subsidiary of Johnson and Johnson makes an intermediate product that bridges EEG and cortical electrode arrays. It is an implant that measures brain waves and also produces electric signal to destroy the large brain waves that are being formed at the beginning of epileptic seizures. Such huge waves can destroy the cortical networks and early detection enables early electrical intervention that can stop the formation of large amplitude electrical-epileptic waves.

Neural prosthetics cover a wide range from retinal implants that give rise to visual perception to the blinds to motor control devices that can use brain signals to control exoskeletons. There is a big gap here from the laboratory to the market, but some of the tools, like EEG controlled wheelchairs, will be available soon. There are diverse applications, like cognitive implants that are capable of influencing and controlling the motion trajectory of the rat.

13.3. 12.3 Robotic tools

Robotic tools are developing very quickly. Artifical hands are highly sophisticated and precise. General exoskeletons can help elderly and motion disable people to move around. Technology is developing quickly and everyday robotic tools are entering a market. Vacuum clearners, small robotic dogs have been on the market for 10 years or so. New tools, lime miniature helicopter, flying machines of different kinds have become widely available, including novel designs like the quadcopter, nit mentioning that robot cars have been riding on the road; Audi, BMW, Toyota, and Google are all developing the next car generation. Legal issues are in the way only. Robotic surgery has made a quantum leap in recent years, see e.g., the Leonardo of Intuitive Surgical, or Amadeus of Titan.

It is hard to predict how much progress will be made in the next 10 years. The major drawback is the weak world economy and not the advancement of the technology.

13.4. 12.4 Tools monitoring the environment

Very harsh developments occurred in the area of Smart Phones. The giant Google has entered the market and we see a huge struggle between FaceBook, Google, Microsoft buying up smaller but sometimes still large or sometimes very quickly developing small companies such as Nokia (large, but it is on the verge), Skype (reasonably sized, but not growing) or Waze (very small, but doubling customers in every 6 months). The key to this new bubble is the market for advertisments, which has limitations; the amount people can spend on goods and can be targeted very quickly via the new social networking, human centered tools like Smart Phones.

These tools have a collection of sensors, starting from webcam and microphone, to gyros, and GPS sensors and also tools for interaction, such as touchscreens, audio and monitor output. The very new generation moves these 'phones' into watches, glasses, or tools that snap-onto glasses. Kinect-like 3D cameras are also on their way. Glasses that monitor the retina and can tell the direction of the gaze simultaneously are also coming very quickly.

In turn, a large part of our daily activities including our environment can be easily monitored and tracked.

13.5. 12.5 Outlook

As a conclusion, I suggest that you read Ray Kurzweil's book ttiled "The Singularity is Near: When Humans Transcend Biology" where Kurzweil claims the following: we are facing a very fast transition to a new era due to the exponential increase in technologies like nanotechnology and thus computers and robotics, genetics technology, and artificial intelligence. This fast transition leads to a technological singularity according to him. At this future point in time technological advances become fast and technological tools become widely available, together with a fast increase in our cognitive capabilities due to human computer confluence.

Similar predictions about the advance of artificial intelligence have been made in the past. They all claimed that AI will surpass human intelligence and typically predicted 10 years. The first prediction was in the fifties (Herbert A. Simon) and now we may safely state that it was false. Another notable instant was in April 1998, when Bill Gates predicted 10 years, which is about over by now. Kurzweil said 15 years in 2005. We are about half way across that period. AI has not developed too much during the last 7 years, but the number of engineers who take part in crowdsourced developments has become huge (see, e.g., the netowrk of Kaggle, or Marinexplore to mention only two orthogonal directions, the first in optimization, the second in high tech sensor sharing).

We finish this outlook by two notes:

Note 1. If our brain is using algorithms for computing the outputs based on its inputs then it is hard to see why human intelligence could not bee reached.

Note 2. If the mammalian brain is built onto the algorithmic principles across species then it is hard to see why it takes so long to overcome the performance of the human brain.

The key might (should) be in mathematics

13.5.1. 12.5.1 Recommender systems

In the last few years we have experienced important breakthroughs in mathematics that relate the NP-hard optimization induced by ℓ_0 norm or the polynomial complexity of the ℓ_1 norm. There is an equivalence between the two norms for certain databases and the conditions of this equivalence seem to be closely matched by databases generated occurring in nature. The two key phrases that appeared in this context are "compressive sampling" and "exact matrix completion".

They are both relevant for recommender systems that can be highly precise since the entropy of our daily routines is very low as shown by Albert-László Barabási and co-workers using mobile phone data. If our 'phone' has access to the unexpected changes in our environment including e.g., the calendars and routes of our family and collaborators then this entropy could be decreased even further. It is intriguing that this way we could increase our control about our life and the seemingly higher freedom gives rise to lower entropy and decreases the uncertainties and stress level of our daily life.

13.5.2. 12.5.2 Big Brother is watching

If our daily life can be predicted with high precision and since the information conveyed to us is based on these predictions we become easy subject of manipulations; targeted advertisements and contents will cover a large part of the world and we will not notice. It is already happening, certain information types can't be found on Google (in realistic times) but hop up easily on Yahoo! It is about time to return to the old technology of internet crawlers to search for information hidden implicitly by search engines.

On the other hand, technology is also available to many people and we have the dilemma: we would like to watch everybody so they can't cause harm and don't want to be watched and predicted since we want to save our privacy and freedom. This dilemma has not been resolved.

14. 13 Decision support tools. Spatial-temporal reasoning.

In several intelligent embedded systems the environment (meant in broad sense) is changing in time and in space; and we are interested in the patterns of that change.

For example in most of the ambient assisted living (AAL) applications we are interested in the activities of the person(s) helped by the system. These complex activities consist of several actions, and the spatial and/or temporal relations among these actions carry a lot of information. Because the habits and regular daily activities should be identified and modeled, the temporal sequence of actions is important. Let us take the breakfast as an example. Even that simple regular daily activity has a lot of components (at least 5-10) and a lot of variations.

- Go to the kitchen
- Take water from the tap
- Boil water make tea (using microwave oven, water boiler, simple oven and a pot etc.)
- Boil water make coffee (using microwave oven, etc.)
- Take some form of bread
- Take meal from the refrigerator
- Prepare cold meal (bread and butter, sausage etc.)
- Prepare hot meal (ham and eggs, hot dog etc.)
- Bring meal to the kitchen table
- Bring meal to the living room
- (Some people eat the breakfast in the kitchen, other eats in the living room etc. Some people watch tv during eating the breakfast other never does it.)
- Eat the meal
- Drink the water, juice or coffee
- Collect the litter

Some actions have some fixed ordered sequence (we first prepare the meal and later eat it), others do not have (sometimes we first drink later eat the bread, in other days the sequence is opposite). Some actions could be repetitive (drinking) others typically are not.

Therefore even for that simple activity a good - and unfortunately complex - model is needed to detect it; and to analyze it whether it is a usual or an atypical (strange or pathological) one.

14.1. 13.1 Temporal reasoning: Allen's interval algebra

One of the possible of temporal reasoning is Allen's interval algebra. It is a calculus for temporal reasoning that was introduced by James F. Allen in 1983. It is used to model the temporal relations among the activities.

The most important attribute of events in Allen's system is the time interval of it. The main purpose of the representation is to find a consistent system of relations among the time intervals of the events. 13 relations were defined among time intervals depending on the relations of the start-times and end-times of them: 6 pairs of relations ("before", "before inverse" etc.) were defined and "equal", which has no different pair.

Notation: $X relation Y = X \odot Y$

Relation ($X \odot Y$)	Time →								
Equal (X *eq* Y)		x	...	x	x				
		y	...	y	y				
Before (X *b* Y)	x	...	x	...					
				...	y	...	y	y	
Before inverse (X *bi* Y)	y	...	y	y	...				
					...	...	x	...	x
Meet (X *m* Y)		x	...	x					
					y	y	y	...	y
Meet inverse (X *mi* Y)		y	...	y					
					x	...	x	x	
Overlap (X *o* Y)		x	x	...	x	x			
					y	y	...	y	
Overlap inverse (X *oi* Y)				x	x	...	x		
		y	...	y	...	y			
During (X *d* Y)				x	x	...	x		
			y	...	y	...	y	...	y
During inverse (X *di* Y)		x	...	x	...	x	...	x	
			y	y	...	y			
Start (X *s* Y)			x	...	x				
			y	y	...	y	...	y	
Start inverse (X *si* Y)			x	...	x	x	x	...	x
			y	...	y	y			
Finish (X *f* Y)					x	...	x		
	y	...	y	...	y	...	y		
Finish inverse (X *fi* Y)		x	...	x	x	x	...	x	
					y	...	y	y	

If there are 3 events - X, Y and Z - then the relations $X \to Y$ and $Y \to Z$ define the possible relations between X and Z.

55. ábra. Relation graph

The question is what do we know about the possible relations between X and Z, if we know that there is relation_1 between X and Y; relation_2 between Y and Z?

Allen gave a composition table, which gives the possibilities having any two simple relations.

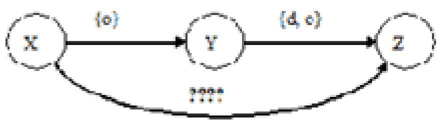
rel_2→ rel_1↓	b	bi	d	di	o	oi
b	b	{all}	{b,d,o,m,s}	b	b	{b,d,o,m,s}
bi	{all}	bi	{bi,d,oi,mi,fi}	bi	{bi,d,oi,mi,fi}	bi
d	B	bi	d	{all}	{b,d,o,m,s}	{bi,d,oi,mi,f}
di	{b,di,o,m,fi}	{bi,di,oi,mi,si}	{d,di,s,si,f,fi,o,oi,eq}	di	{di,oi,fi}	{di,oi,si}
o	b	{bi,di,oi,mi,si}	{d,o,s}	{b,di,o,m,fi}	{b,o,m}	{d,di,o,oi,s,si,f,fi,eq}
oi	{b,di,o,m,fi}	bi	{d,oi,f}	{bi,di,oi,mi,si}	{d,di,o,oi,s,si,f,fi,eq}	{bi,oi,mi}
m	b	{bi,di,oi,mi,si}	{d,o,s}	b	b	{d,o,s}
mi	{b,di,o,m,fi}	bi	{d,oi,f}	bi	{d,oi,f}	bi
s	b	bi	d	{b,di,o,m,fi}	{b,o,m}	{d,oi,f}
si	{b,di,o,m,fi}	bi	{d,oi,f}	di	{di,o,fi}	oi
f	b	bi	d	{bi,di,oi,mi,si}	{d,o,s}	{bi,oi,mi}
fi	b	{bi,di,oi,mi,si}	{d,o,s}	di	o	{di,oi,s}

Table 13.1 Part of Allen's composition table
(in the left column relation1 is given in the uppermost row relation_2 is given, in the boxes there is the set of the resulting relations)

If there is more than one possible relation between two events, the union of the sets gives the result.

Example 13.1

X → Z: {o→ d} U {o→ d}={d,o,s} U {b,o,m}= {b,d,o,m,s}



56. ábra. Using the composition rules

The situation is more complex if there are a lot of events, and we know some relations among them. We model the events by a graph, the vertices are labeled by the vents, the edges are labeled by the relations.

A consistent singleton labeling of the graph is a labeling where it is possible to map the intervals to the time scale and there is only one single relation between any two events. In minimal labeling the edges are labeled by a set of relations, the elements of the set are all part of a singleton labeling.

Given a labeled graph one of the most important questions is whether there is contradiction between any two relations or not. Three possibilities are considered:

- Exact solution of the problem is an exponential problem.
- Easier special cases limiting the expressive power of the representation.
- Approximation: every triangles of the graph is checked to be consistent. If there is any discrepancy, the label set of one of the edges is modified. At the end every triangle will be consistent but the whole graph could remain inconsistent.

Because exact solution is an NP-complete problem, the approximate solutions are very important.

Pseudo code of an approximation algorithm, the path-consistency algorithm is given. This algorithm checks all the triangles in the graph, whether there is contradiction in it or not. It is an approximation, because if all the triangles are contradiction free, there could remain contradiction in larger loops.

```
function Path-Consistency(C,n)
        /* C matrix is the initial label set nxn matrix */
        /* n is the number of vertices*/
        /*C_jk is the relation label set attached to the edge connecting
        /* vertices j and k*/
        Q←{(j,k)|1 ≤ j < k ≤ n}      /* all pairs of vertex indices */
        while Q is not empty do
                take one pair of indices from Q: (j,k)
                for i=1 to n
                        if (i=j) or (i=k)
                        then
                                do nothing
                        else
                                temp ← C_ji ∩ (C_jk ⊗ C_ki)
                                if temp ≠ C_ji
                                            C_ji ← temp
                                            C_ij ← inverse(temp)
                                             Q ← Q ∪ {(j,i)}
                                     endif
                                temp ← C_ik ∩ (C_ij ⊗ C_jk)
                                if temp ≠ C_ik
                                            C_ik ← temp
                                            C_ki ← inverse(temp)
                                             Q ← Q ∪ {(i,k)}
                                endif
                        endif /* if (i=j) or (i=k) - else */
            endfor
      endwhile
      return C
endfunc
```

14.2. 13.2 Spatial reasoning

Humans are usually much better in dealing with spatial information than computers. There are two types of problems: the uncertainty and the incompleteness of the information given or measured. An example for the first one is the uncertainty of the spatial measurements of some mobile robots in order to determine its position. The incompleteness occurs usually in human-computer systems, where some information is given using natural language. For example: "the shop is on the left half way to the square". This information is uncertain as well (half way is not a strict quantity or location), but it is incomplete as well, if the uncertainty is solved somehow, "on the left" is satisfied by infinite number of locations. In several scenarios these uncertain and incomplete pieces of information describe only local, relative position of the entities, but we are interested in the global positioning of one or more entities. (Find something on the map, which is not far from this, left from that etc.)

Spatial reasoning means to represent knowledge of spatial entities, of spatial relations, and combining the typically uncertain and incomplete pieces of information.

There are several ways of spatial reasoning. The errors could be modeled using two or three dimensional probabilistic distributions. E.g. a stochastic map gives the spatial location of an object with respect to the world reference frame, and a covariance matrix describing the uncertainty of each location information. Active sensing allows updating using Markovian error estimation, Kalman filtering etc.

Incompleteness is usually dealt with qualitative representations and some type of formal logic. If quantification is needed, fuzzy logic could be a bridge between the qualitative and quantitative world.

There are some spatial methods inspired by the temporal interval methods (shown in 13.1). The spatial relations among the objects are projected to two orthogonal axes, and interval based representation is used on each axis.

The reasoning based on incomplete information could be solved by probabilistic representation and methods as well. Consider the following demonstrative example.

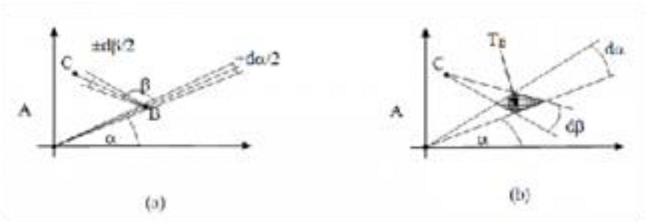
Example 13.2 Let C be an unknown point. Let C be in the direction β from a point B, this point being itself in the direction α with respect to a reference point A. What can we say about the position of point C with respect to A?

(For example: to find the public toilet go along the North boulevard, when you are close to the church, you can see the public toilet looking right a bit back...)

We can choose polar coordinates for location representation, and point A to be origin of the coordinate system. Possible locations of each P point are represented using a probability distribution:

$$f_L(r,\theta) = \lim_{dr,d\theta \to 0} \frac{P(L : r_L \in [r \pm dr/2], \theta_L \in [\theta \pm d\theta/2])}{dr \cdot d\theta}$$

The original problem could be shown as in Fig. 13.2a



57. ábra. Demonstrative example for spatial reasoning (a), redrawn to show the info combination (b)

If we have information (known or estimated distribution information) about B and C, then the information could be combined in the way shown in Fig.57 (b):

$$P(C,\alpha,\beta) = P(C) \int_{T_B} P(B)dB$$

where the domain TB is limited by four lines with equations:

$$r = r_C \frac{\sin(\theta_C - [\beta - \pi \pm d\beta/2])}{\sin(\theta - [\beta - \pi \pm d\beta/2])}$$

$$\theta = \alpha \pm d\alpha/2$$

If there is only such information that C is left from the AB line, it could be modeled using uniform distribution. If B is about the middle of an AX segment of the line having angular coordinate α , then it could be modeled using a Gaussian distribution etc.

14.3. 13.3 Application of spatiotemporal information

One important application area of spatiotemporal reasoning is the analysis of human behavior. Of course there are several problems even within this field: human behavior in the working place, the movement of the crowd in public places, etc. Nowadays the analysis of human behavior at home is of growing importance, because there are lots of elderly people with health risks living alone due to the demographic changes. In this field the application of the temporal or spatial reasoning is not straightforward and simple.

For example the theoretical method (reasoning based on time intervals) shown in 13.1 for temporal reasoning has a major drawback; it does not take into account start time (absolute time) and the duration of the event. In human behavior analysis these are very important factors: if you go to the kitchen in the morning it is probably breakfast, if you go in the evening it is probably dinner. If the person goes for the bathroom for 2 minutes he/she probably washed his/her hands. If the person went there for 30 minutes he/she probably took a shower or a bath. In this problem the location is usually means simply an event, or one of some events. In the bathroom there could be shower, bath, washing etc. depending on the timing and duration. In the kitchen it could be breakfast, lunch, dinner etc. depending again on the timing and duration.

Human activity patterns were analyzed using a home sensor network. Every event is characterized by a triplet: it has a location, a start time, and time duration. Location is given by the room where the active motion sensor is deployed. During the analysis complex behavioral patterns, episodes are looked for. Especially frequent episodes are important, which can characterize the life of the person. (Of course there are random, unique episodes in our life, but these do not characterize us, or at least hard to characterize based on these events. The way we take our breakfast, lunch and dinner, the length and regularity of our sleeping periods, the toilet usage etc. could be important in estimating our health state.)

For example a typical night/morning activity pattern is the following (the time duration is in minutes):

<{Bed",11:00pm",300min},{Bath",4:00am",5min},{Bed",4:06",300min},...

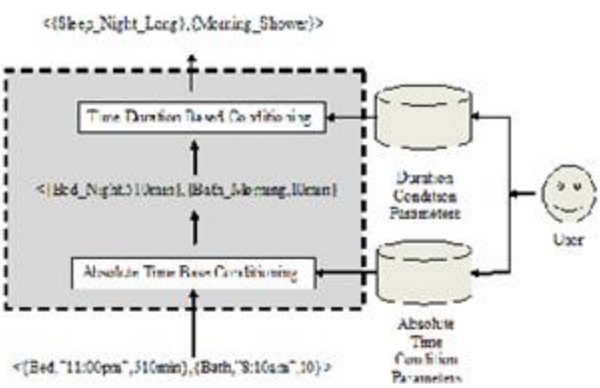
{Bath",9:05am",10min},{Kitchen",9:15am",25min}>

In this context an episode is a cluster of triplets (events), which occur frequently. The problem is that the same episode could run its course with slightly different time and duration data. For example the following episode is the same as the previous one.

<{Bed",10:45pm",275min},{Bath",3:20am",3min},{Bed",3:23am",320min},...

{Kitchen",8:40am",17min}>

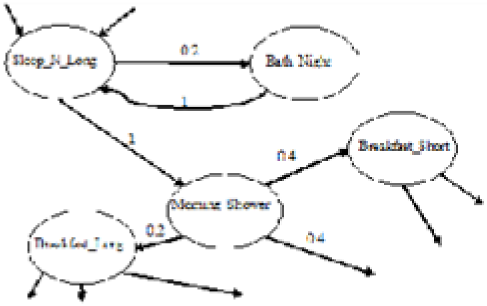
The solution is that the location and the absolute time (start time) is encoded in the event, the time and the duration are roughly quantized to solve the uncertainty of the timing mentioned above. For example if the location is Bed and the start time is between 10:00pm and 12:00pm, the duration is at least 60min, but not more than 600min then the event is encoded as Sleep_N_Long (Sleep, Night, Long). The basic coding system suggested is shown in Fig. 58.



58. ábra. Temporal abstraction layer suggested in [Lymb2008]

Using these encoded events a state-transition model could be learnt, something similar to the one given in figure 59.

<{Bed",11:00pm",510min},{Bath",8:10am",10}>



59. ábra. Part of the state transition model characterizing the daily activities of a human

In Figure 59 a part of the state transition model is shown. The edges are labeled by the approximate probabilities (relative frequencies) learnt. A model like that could be used by checking probability of the actual behavior. If it is strange (the probability of it is low), possibly something happened, for example the health state of the person has changed. (E.g. there are more bath visits during the night than usual.)

15. 14 Activity prediction, recognition. Detecting abnormal activities or states.

15.1. 14.1 Recognizing abnormal states from time series mining

In plenty of embedded intelligent system applications (e.g. AAL, and other intelligent spaces) the working regime of the information collecting system can be characterized as round-the-clock, 7-days-a-week, with multitude of sensory channels. Even if the computation of the context involves abstraction, reduction, resume making and in general dimension and volume reduction, lengthy records are rather a rule than an exception. It means that the problem of data mining, or rather time series mining is inherent to such applications.

What is not so clear for the first glance is that the resources to spend on mining in such application domain could be spare and the mining task is made exceptionally difficult by the lack of good a priori models. It is easy to recognize e.g. in a record those parts with a speech signal and the noisy speechless empty segments, but it is much more difficult to identify the behavioral aspects of the human user in sensory data only partially suited for this purpose, and where the information appears only implicitly.

15.1.1. 14.1.1 Time series mining tasks

In the following we review the main areas of signal processing where effective means to mine extensive signal repositories is essential:

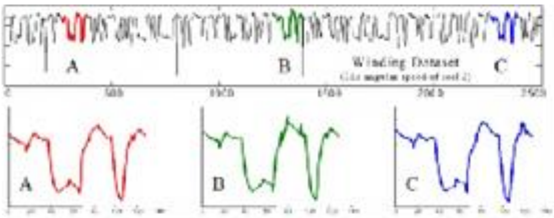
- Indexing: Given a query time series $x(t)$, and some similarity/ dissimilarity measure $D(x,y)$, the problem is to find the most similar time series $y(t)$ in the database of signal records.
- Clustering: Looking for "natural" groupings of the time series in the database under some similarity/ dissimilarity measure $D(x,y)$.
- Classification: Given an unlabeled time series $x(t)$, assign it to one of two or more predefined signal classes.

- Summary: Given a very lengthy time series $x(t)$ containing N data points, create an approximation of $x(t)$ which retains its essential features but contains much less data points and can be outputted and globally inspected fitting e.g. a single page or computer screen.

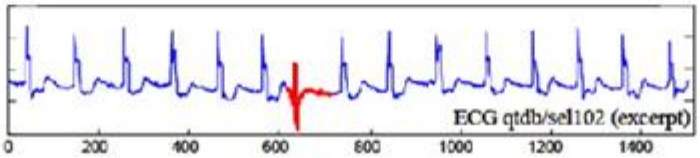
- Anomaly detection: Given a time series $x(t)$, and some model of "normal" behavior, find all sections of $x(t)$ which contain anomalies or "surprising/ interesting/ unexpected/ novel" behavior (see Fig. 2).

- Motif detection: Given a time series $x(t)$, find out in $x(t)$ those segments which are very close (similar) copies of each other (recurring motif) (see Fig.1).

It can be seen that the majority of time series mining tasks requires some similarity measure based on some mathematical distance to be effectively computed.



60. ábra. Time series from an electrical motor showing a motif - self-similar segments contrasting with the whole of the parent signal. (from slide 43, Lin, J., Keogh, E., Lonardi, S. and Chiu, B., A Symbolic Representation of Time Series, with Implications for Streaming Algorithms. Proc. 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, San Diego, June 13, 2003, http://www.cs.ucr.edu/eamonn/SAX.ppt)



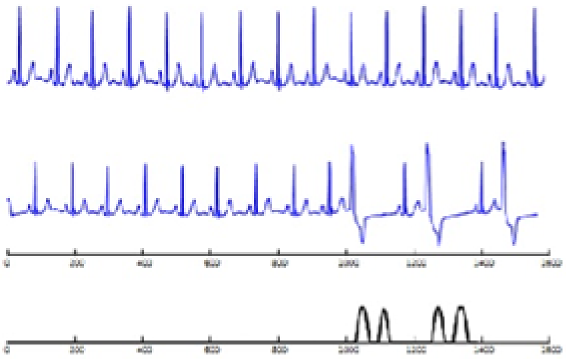
61. ábra. Time series anomaly found in a record of ECG (E. Keogh, J. Lin, A. Fu, HOT SAX: Efficiently Finding the Most Unusual Time Series Subsequence, Fifth IEEE Int. Conf. on Data Mining, 27-30 Nov. 2005, Houston).

15.1.2. 14.1.2 Defining anomalies

In the AAL setting time series data mining is needed for two purposes. Clustering, classification, and motif detection are needed to establish repeatability - experimental and heuristic behaviour model of the human inhabitant, the human "user" of the ambient intelligent space. The opposite is the problem to spot the divergence from that model, the anomaly in the human behaviour, indicating perhaps that something grossly wrong is happening. Such discrepancy is the most important information component in the collected context. It is that information and the moment when the embedded system must draw on its resources, exercise its intelligence and decide with whom how to interact, and perhaps what to affect in the physical space. The discrepancy can be without consequence, but it can be an indication of an alarming situation demanding immediate attention of professional people (e.g. the inhabitant of the ambient space - an elderly - fell down and cannot get up by himself).

Ideally we would like the system to process time records in such a way that non-anomalous time segments, albeit differing, should not even raise over the threshold of the system's attention. In contrary, anomalous segments, even if differing slightly from the others, should lead to an unambiguous indication of problems.

The problem of anomaly is that of the model. If we possess a good (mathematical) model of the standard behaviour, we can derive from it the model of the anomalous behaviour, then we can compute derived features observable in the abstracted or fused signals, where the problem of dimension and volume is less strident. As we usually do not have a good abstract model of the anomaly (what could be a good abstract and identifiable phenomenon in the sensory data of a human agent e.g. tumbling down, misusing the microwave, or not taking the required medicine?), we must remain at the lowest levels of the context, working with rough sensory data, where the problem of dimension and volume can be paramount.

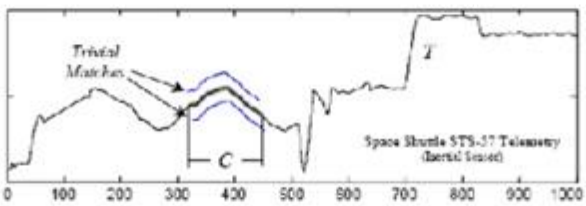


62. ábra. An ideal behavior: reacting only to the anomalous segments in the data (from slide 36, Lin, J., Keogh, E., Lonardi, S. and Chiu, B., A Symbolic Representation of Time Series, with Implications for Streaming Algorithms. Proc. 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, San Diego, June 13, 2003, http://www.cs.ucr.edu/eamonn/SAX.ppt)

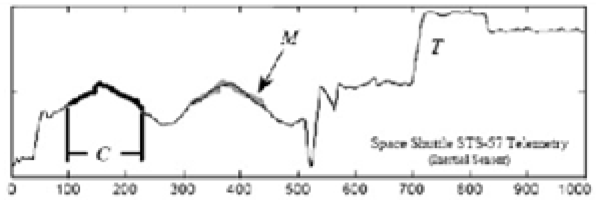
The only fact we can cling to is the recognition that anomaly means in general something differing the most from any kind of signal appearance considered normal behaviour. An anomaly cannot be similar to anything good. Such signal segment we will call a discord. To this purpose we must define in the signal a match, i.e. the relation of two signal segments based on some similarity (distance) definition. Matching segments are by definition very closed, very similar to each other.

In looking for matching segments we must exclude segments shifted by one or some sample points. If the signals are relatively smooth, but noisy, the best match will be always a one sample shift. We call such matches trivial and will get rid of them asking for "real" matches placed well far off from each other.

Definition 1. Two segments are non-self match to each other, if they are matching but are placed at least a segment length apart from each other.



63. ábra. Illustration of the definition of a trivial match. (from B. Chiu, E. Keogh, S. Lonardi, Probabilistic discovery of time series motifs, 9th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, KDD'03, 2003, pp. 493-498.)



64. ábra. Illustration of the definition of a non-self match. (from B. Chiu, E. Keogh, S. Lonardi, Probabilistic discovery of time series motifs, 9th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, KDD'03, 2003, pp. 493-498.)

Definition 2. A segment D of length n is said to be the discord of the signal record if D has the largest distance to its nearest non-self match. That is, for every C segment in the record, non-self match MD of D, and non-self match MC of C, min$(Dist(D, MD)) >$ min$(Dist(C, MC))$. The property of being a discord depends on the segment length, so some a priori expectation should be always thrown here in.

Definition 3. A segment D of length n beginning at position p is the $\mathbf{K^{th}}$-discord if D has the K^{th} largest distance to its nearest non-self match, with no overlapping region to the i^{th} discord beginning at position pi, for all $1 \leq i < K$. That is, $|p - pi| \geq n$.

For comparison let us mention also:

Definition 4. The most significant motif in the signal record is the segment $C1$ that has highest count of non-trivial matches (ties are broken by choosing the motif whose matches have the lower variance).

15.1.3. 14.1.3 Founding anomalies, from using brute-force to heuristic search

When we settle on how long is the anomalous segment we try to identify, such a segment can be always found by a brute force search along the signal. Whether it is a real anomaly or not must be judge later based on distance data to other segments in the signal. What we must do is to run a window of the segment length along the signal record, find the non-self match to it, store its distance information and go on. Then we must identify the segment with the most distant non-self match. So the algorithm should run somehow like this:

```
function [distance, location ]= brute_force(t, n)
  best_so_far_distance = 0
  best_so_far_location = NaN
  for p = 1 to |T| - n + 1                          %% Begin Outer Loop
  nearest_neighbor_distance = infinity
    for q = 1 to |T| - n + 1                        %% Begin Inner Loop
      if |p - q| ≥ n                                %% non-self match?
        if Dist (t_p,...,t_p+n-1, t_q,...,t_q+n-1) < nearest_neighbor_distance
                                                    %% (*)
            nearest_neighbor_dist = Dist (t_p,...,t_p+n-1, t_q,...,t_q+n-1)
        end
      end %% End non-self match test
    end    %% End Inner Loop
    if nearest_neighbor_distance > best_so_far_distance
            best_so_far_distance = nearest_neighbor_distance
            best_so_far_location = p
    end
  end  %% End Outer Loop
  return[best_so_far_distance, best_so_far_location]
```

where T is the signal record, n is the length of the investigated segment, and $Dist(t_p, \ldots, t_{p+n-1}, t_q, \ldots, t_{q+n-1})$ is the distance value between two segments from T, both of length n, starting from p and q sample points respectively.

Studying the brute-force approach we can made two important observations, which shortly will lead to a much more effective heuristic search based on early pruning.

Observation 1: When running in the inner loop, we don't actually need to know the true nearest neighbor to the current candidate. When we find any segment that is closer to the current candidate than the best_so_far_distance, we can break off from the inner loop, being sure that the current candidate cannot be the expected discord.

Observation 2: The effectiveness of the Observation 1 depends on the order of the candidates (discords to be) proposed in the outer loop, and on the order of segment evaluation in the inner loop in the attempt to find a sequence that permits the early break off.

Let us assume that the dull outer and inner cycle gives place to a well informed heuristics. Imagine that an oracle reveals the best possible orderings. For the outer loop, the segments are sorted by descending order of the non-self distance to their nearest neighbor, that way the true discord will be the first object examined. For the inner loop, the segments are sorted in ascending order of distance to the current candidate. In this situation the first invocation of the inner loop will complete the computation. All subsequent invocations of the inner loop will be abandoned during the very first iteration. The time complexity is thus just $O(m)$ (where m is the length of the time series, $m \gg n$).

Observation 3: In the outer loop, a considerable speedup can be achieved with less perfect ordering. We really only need that among the first few segments being examined, at least one has a large distance to its nearest neighbor. That way the best_so_far_distance variable will acquire a large value early on, which will make the conditional test (*) be true more often, allowing more early terminations of the inner loop.

Observation 4: In the inner loop, a considerable speedup can be achieved with less perfect ordering. We really only need that among the first few segments being examined, at least one has a distance to the candidate segment (being considered for discord) that is less than the current value of the best_so_far_distance variable. This is sufficient for an early termination of the inner loop.

```
function [distance, location]= Heuristic_Search(T, n, Outer, Inner)
  best_so_far_distance = 0
  best_so_far_location = NaN
  for each p in T ordered by heuristic Outer
                                    %% Begin Outer Loop
    nearest_neighbor_distance = infinity
    for each q in T ordered by heuristic Inner
                                    %% Begin Inner Loop
      if |P-Q|≥ N                   %% non-self match?
        if dist (Tp,…,Tp+N-1, TQ,…,TQ+N-1) < best_so_far_distance
          break                     %% Break out of Inner Loop
        end
        if dist (tp,…,tp+n-1, tq,…,tq+n-1) < nearest_neighbor_distance
         nearest_neighbor_distance = dist(tp,…,tp+n-1, tq,…,tq+n-1)
        end
      end                           %% end non-self match test
    end                                 %% end inner loop
    if nearest_neighbor_distance > best_so_far_distance
      best_so_far_distance = nearest_neighbor_distance
      best_so_far_location = p
    end
  end                               %% end outer loop
  return[best_so_far_distance, best_so_far_location]
```

The question now is how to design heuristics possessing the required properties, yet computable easily. To this aim we will investigate shortly how time series can be represented, then we will choose an equivalent representation (to the original time series) where the design of the heuristics will be straightforward.

Many high level representations have been proposed for data mining see Fig. 62. Among them symbolic representations were also considered, because such representations would potentiality allow the usage of the abundance of the text processing algorithms. Many symbolic representations have been already introduced, however to common problem is that the dimensionality of the symbolic representation is the same as the original data (data mining algorithms are sensitive to the dimensionality) and that distance measures defined on the symbolic series have little correlation with distance measures defined on the original time series.

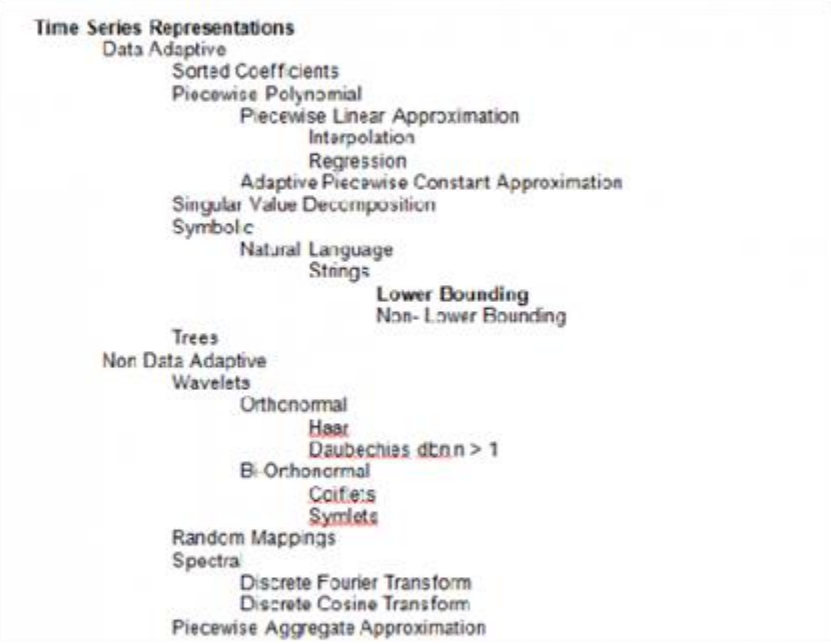


Figure 14.5.b Time series representations in time series data mining.

In the following a suitable symbolic representation will be introduced, so called SAX, belonging to the 'lower bounding' slot in Fig. 3 which will be space and time efficient and will produce correct results when mined instead of the parent numerical representation. This will be an ideal candidate to formulate inner and outer heuristics (working well, computed and processed easily).

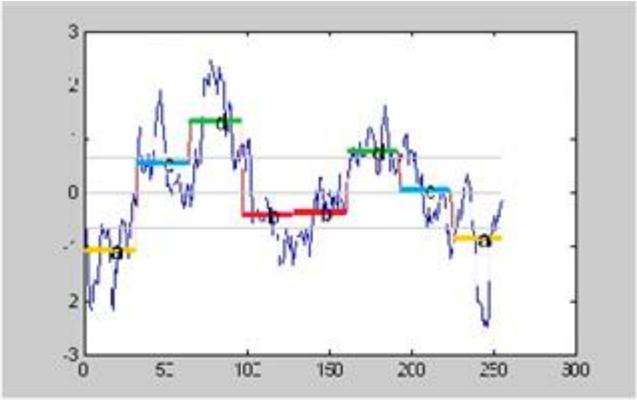
15.1.4. 14.1.4 SAX - Symbolic Aggregate approXimation

SAX method allows a time series of arbitrary length n to be reduced to a string of arbitrary length w , (typically $w \ll n$). The alphabet size for the string is also an arbitrary integer $a, a > 2$.

The first step is to transform the original time series into so called Piecewise Aggregate Approximation (PAA) representation, which is a segment-based approximation with constant levels computed as the mean value of the signal in the given segment.

The second step involves transition to symbolic representation where every segment is assigned a letter from a finite alphabet based on the principle of the uniformly probable distribution of the alphabet symbols. To that aim the empirical distribution of the original time series is computed or approximated and threshold levels (called 'breakpoints') delimiting amplitude regions of equal probability, one region for each symbol from the alphabet, are computed. If a segment level in PAA falls into a particular amplitude region, a symbol labeling this region will be used to code the segment.

As a result a time series of length n will be transform into a symbolic series of length w , a 'word' built from symbols from the alphabet, see Fig. 63.

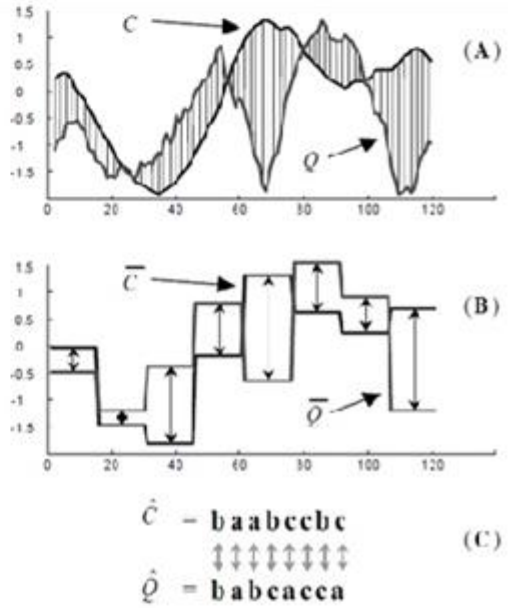


65. ábra. Transformation of the analytic (numerical) time series, via the PAA representation, into the concise symbolic SAX series 'acdbbdca'. (http://www.cs.ucr.edu/eamonn/SAX.htm)

The most important issue in transforming representations is the behaviour of the distance measures. Every representation has a distance measure defined with different math (pertinent to the representation). To be able to solve a task posed in one representation by more efficient means in another representation, requires a well defined relation between the distance measures. Otherwise changing representations serves no purpose.

Distances among time series segments in the original time series and the PAA representation are defined as the Euclidean distance on the real line. Distance under SAX representation is based on the distance of the letters, which is based on how far are from each other the amplitude regions defining them. The essential observation is that in the original - PAA- SAX representation chain, with the mentioned distance measures, the distances are lower bounding each other, i.e.:

$$Dist(x, y) \geq Dist(x_{PAA}, y_{PAA}) \geq Dist(x_{SAX}, y_{SAX})$$



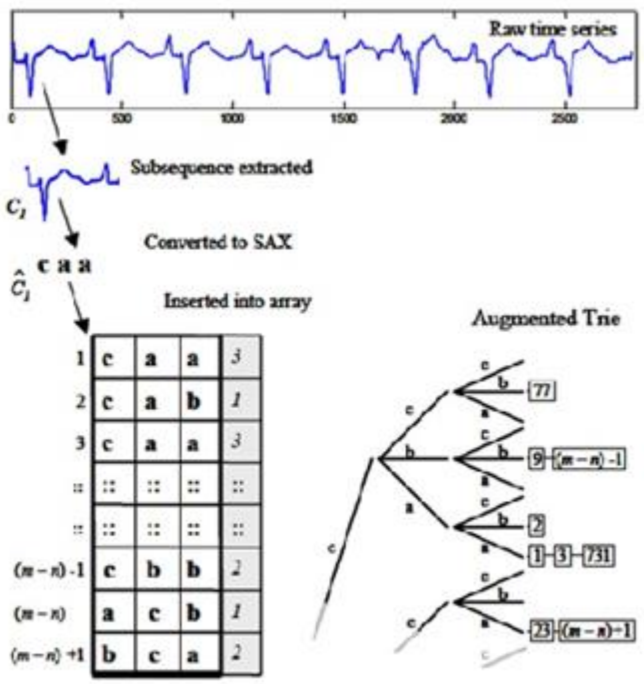
66. ábra. **A)** The Euclidean distance between two time series is the square root of the sum of the squared differences of the corresponding amplitudes. **B)** The distance measure in the PAA representation is the square root of the sum of the squared differences of the corresponding levels, normalized by the compression rate. **C)** The distance between two SAX representations is the sum of the squared symbol distances, taking the square root and normalizing by the compression rate. (Lin, J., Keogh, E., Lonardi, S. and Chiu, B., A Symbolic Representation of Time Series, with Implications for Streaming Algorithms, 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, San Diego, June 13, 2003)

15.1.5. 14.1.5 Approximating the search heuristics

The SAX representation introduced above can be used to create from the original lengthy time series a clever symbolic database, which in turn can be used to efficiently computing the heuristics mentioned in the speedup algorithm. To that aim we must run a window along the signal sample-by-sample, SAX-transform each cut segment into a SAX-word, and:

1. build an array of SAX-words, sample-by-sample, and associate array words with their frequency of appearance in the time series,

2. build a trie (word tree) from the SAX-words, and associate its leaves with the position lists, where a given SAX-word appears in the original time series, see Fig. 65.



67. ábra. Data structures required to compute the Inner and Outer heuristics. An array of SAX-words with the last column containing a count of how often each word occurs in the array (left). A fragment of an trie with leaves that contain a list of all array indices (i.e. signal segments) that map to that particular branch (right). (E. Keogh, J. Lin, A. Fu, HOT SAX: Efficiently Finding the Most Unusual Time Series Subsequence, Fifth IEEE Int. Conf. on Data Mining, 27-30 Nov. 2005, Houston)

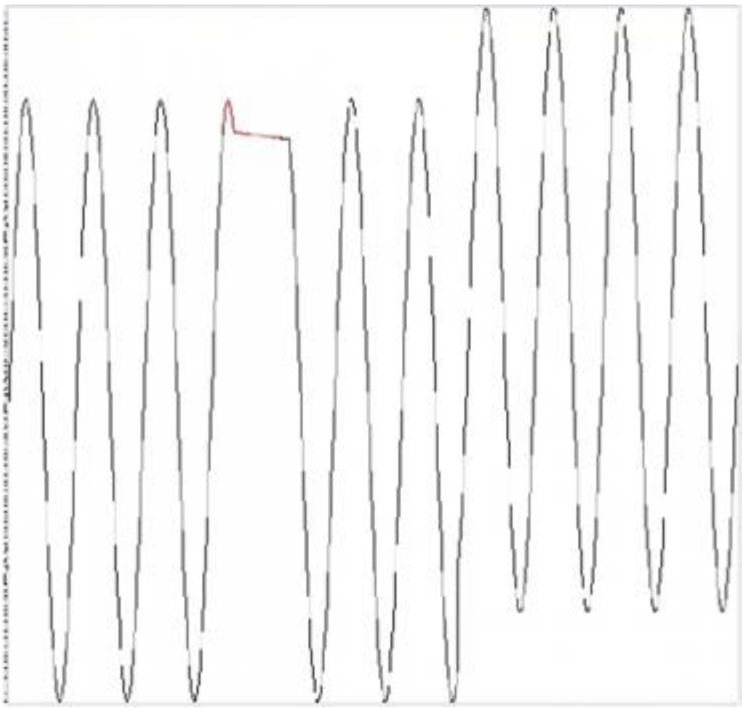
Both data structures can be created in time and space linear in the length of the original time series.

To compute Outer heuristic we take the rightmost column of the array looking for the smallest count (virtually always 1). The indices of all SAX-words that occur so many times are given to the outer loop to search over first. After the outer loop has exhausted this set of candidates, the rest of the candidates are visited in random order.

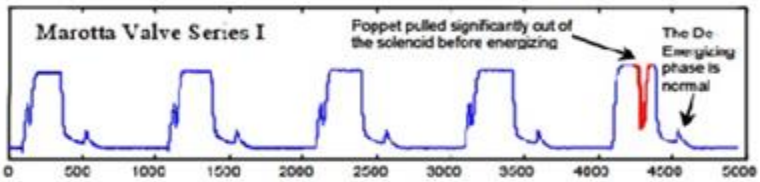
The explanation is simple. Unusual segments are likely to result in unique or at least rare SAX-words. By evaluating the candidate segments that led to unique or rare SAX words early in the outer loop, there is an excellent chance of providing a large value to the best_so_far_distance variable early on, see Observation 3.

Inner heuristic can also profit from the data structures in Fig. 6. When an ith candidate segment is first considered in the outer loop, we look up its SAX-word examining the ith word in the array. Then we retrieve from the trie the list at this particular word and order the first items in the inner loop acc. to it. The rest of the segments is visited in random order.

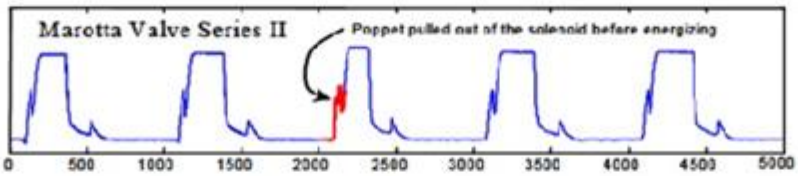
Here the explanation is also simple. Segments that lead to the same SAX-words as the candidate segment are likely to be highly similar. By Observation 4, it is enough to find one such segment that is similar enough in order to terminate the inner loop.



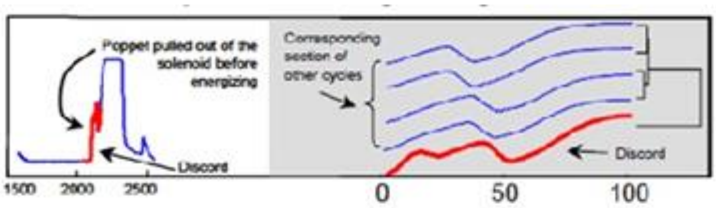
68. ábra. A discord in artificial test data.



69. ábra. Annotated Marotta Valve time series (ISS). The discord discovered exactly corresponds with the expert's annotation (E. Keogh, J. Lin, A. Fu, HOT SAX: Efficiently Finding the Most Unusual Time Series Subsequence, Fifth IEEE Int. Conf. on Data Mining, 27-30 Nov. 2005, Houston)



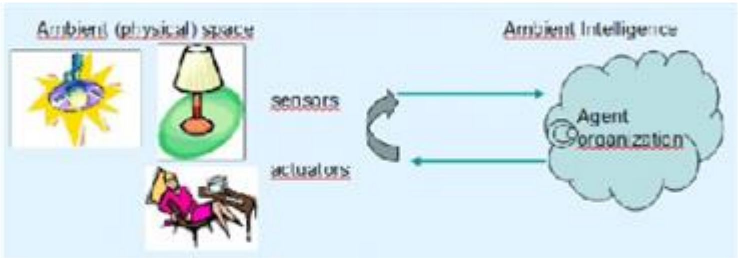
70. ábra. Annotated Marotta Valve time series. While the discord discovered exactly corresponds with the expert's annotation, it is difficult to see why. (E. Keogh, J. Lin, A. Fu, HOT SAX: Efficiently Finding the Most Unusual Time Series Subsequence, Fifth IEEE Int. Conf. on Data Mining, 27-30 Nov. 2005, Houston)



71. ábra. (Left) A part Fig. 9with the discord found. (Right) Discord blown up, and four other corresponding sections form the normal cycles. The engineer noticed the unusual „double hump" which flagged the problem. (E. Keogh, J. Lin, A. Fu, HOT SAX: Efficiently Finding the Most Unusual Time Series Subsequence, Fifth IEEE Int. Conf. on Data Mining, 27-30 Nov. 2005, Houston, also from slide 36, SAX time series/Shape tutorial, http://www.cs.ucr.edu/eamonn/SIGKDD_2007.ppt).

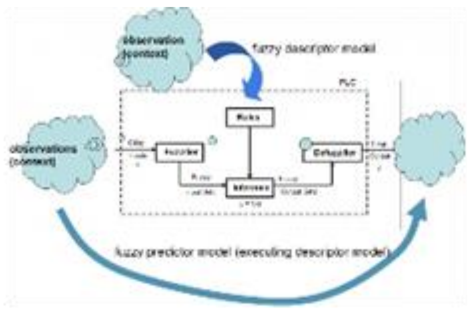
15.2. 14.2 The problem of ambient control - how to learn control the space from examples

We will study now the problem how an embedded intelligent system can learn the control from the human. Originally it is the human inhabitant who controls the environment. Ambient intelligence can observe his actions, relate them to the (also) observed environmental conditions (and other components of the context) and after a while (when enough information is collected) can formulate control rules which when employed, will satisfy and free the human from some mundane tasks.



72. ábra. Ambient control learning problem. Ambient intelligence observes the space and the human actions and generate rules - a model of the human from control point of view.

For the learning mechanism we chose the fuzzy logic controllers as being easy to handle and well adaptable to adverse learning conditions. Learning fuzzy controller won't be easy because due to the character of the problem the learning must happen on-line, with rare data, and in a non stationary environment.



73. ábra. Fuzzy controller as a model module for the ambient control system. Fuzzy controller fuses together the predictive and the descriptive model of the acting human.

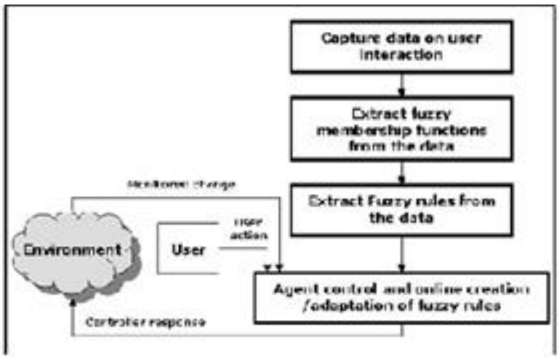
The approach presented in the following was developed for the embedded system governing the Essex intelligent Dormitory (iDorm) (F. Doctor, H. Hagras, and V. Callaghan, A Fuzzy Embedded Agent-Based Approach for Realizing Ambient Intelligence in Intelligent Inhabited Environments, IEEE Trans. on Syst., Man, Cybern.-Part A: Systems and Humans , Vol. 35, No. 1, Jan 2005, pp. 55-65.). In that environment the living conditions of the human inhabitant were controlled by the heater, cooling ventilator, 4 small lamps of settable light intensity, bedside lamp, PC word processor application, and PC media entertainment application. This in turn was sensed with in-door light sensor, out-door light sensor, in-door temperature sensor, out-door temperature sensor, humidity sensor, chair pressure sensor, bed pressure sensor, presence sensor, and phone status sensor.

15.2.1. 14.2.1 Adaptive Online Fuzzy Inference System (AOFIS)

The learn-and-control mechanism is composed from a one-shot steps covering the monitoring of the (behaviour of the) inhabitant by collecting the input/ output data (sensory data/ actuator setting). Then comes the identification of the fuzzy membership functions. As the physical character and range of the collected signal is known from the intelligent space set-up, the universes for the membership functions are also set. The question is how many of them along a given universe and where to place them. The answer to it is hidden in the collected signals; it must be only properly mined. With membership functions ready the next step is to identify fuzzy rules. We can prescribe their format, but we must find out which particular (sensor) fuzzy set are needed as a premises for a particular (actuator) fuzzy set in the consequence. This we also do by properly mining the collected information.

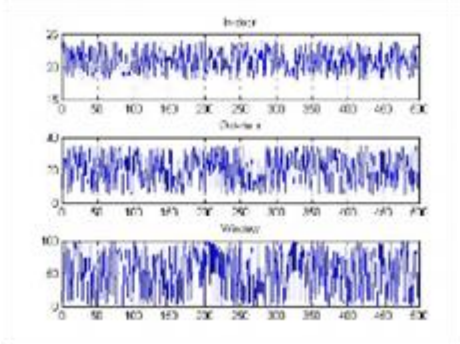
Equipped with rules the monitoring system enters the control loop, where based on the actual sensory data it makes an attempt to predict the likely reaction of the inhabitant and to perform it on his or her account. if the prediction if approximately correct, then the human inhabitant will never want to override the automatic setting. If however the circumstances changed (got into a mood, the weather changed unpredictably, somebody else came to spend time in this spaces, etc.) it may happen that the automated setting won't satisfy the user, who will reset the actuators to his or her liking.

The monitoring system, when overridden, should memorize such data, as essential for further adaptation. When enough body of this data is already collected it is time to reconsider the rules, to adapt them by modifying the reference to the particular fuzzy set, and in the long run, even to modify the membership functions themselves. After this the system is tuned anew and can take over the control. Until some new change in the conditions appears and the adaptation resumes.



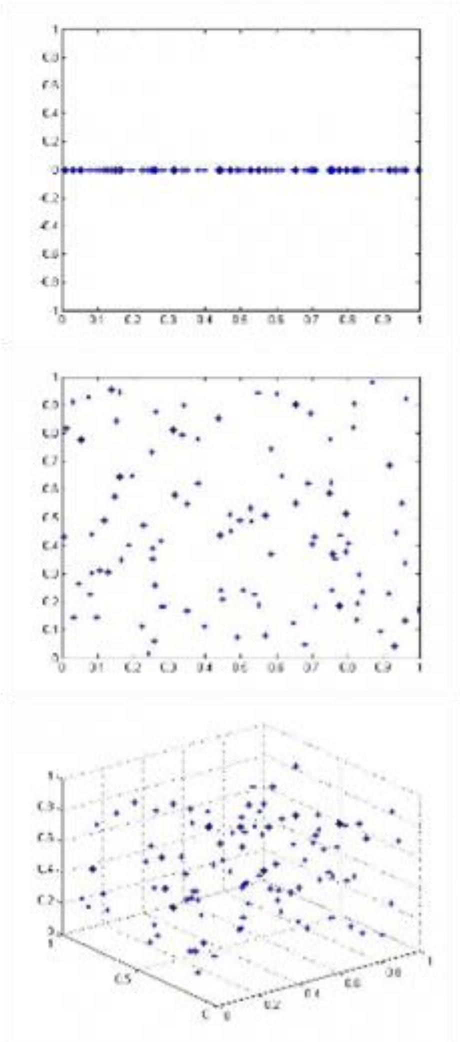
74. ábra. One-shot time series mining and learning control loop. (F. Doctor, H. Hagras, and V. Callaghan, A Fuzzy Embedded Agent-Based Approach for Realizing Ambient Intelligence in Intelligent Inhabited Environments, IEEE Trans. on Syst., Man, Cybern.-Part A: Systems and Humans , Vol. 35, No. 1, Jan 2005, pp. 55-65.)

In the following the show the working of the algorithm on a simplified example of the "window opening" control based on in-door and out-door temperature measurements. It is summer time, so the out-door temperature range (universe) is 0 - 40 ◦ C, in-door is a temperate 15 - 25 ◦ C, and the window can be 0 - 100 % open. The data collected (from a simulator of human behaviour) can be seen in Fig. 75.



75. ábra. Measured input/ output data.

One of the first problem to tackle is so called "curse of dimensionality" meaning that the more dimensions we have, the more data we must collect to cover the whole space with the required resolution. In Fig. 76 we see 100 data points in 1, 2, and 3 dimensions, and we can clearly observe, how the data cloud gets sparser and sparser.



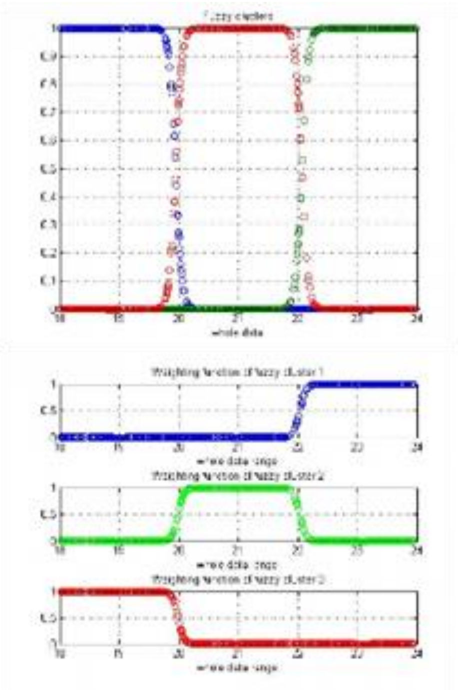
76. ábra. abc. The curse of dimensionality. 100 data points dense in 1 dim, but getting sparser in 2, or 3 dimensions.

Next step is the identification of the number and position of the fuzzy membership functions (over three universes mentioned earlier). It can be done in an ad hoc manner, but we can make an attempt to be objective and let the data talk. To this aim we propose (1) to fuzzy cluster the data in the respective dimension (Fuzzy C-Means - FCM), and (2) to use the cluster weighting function to generate the membership function.

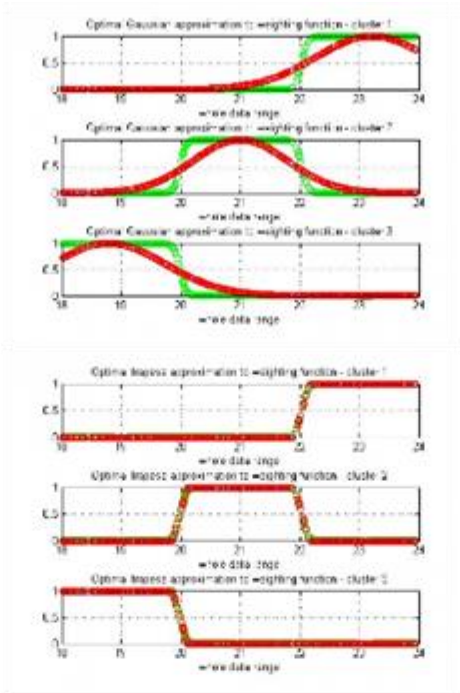
77. ábra. Fuzzy clustering separates the data points in a „soft" way. A given data point can belong to a number of clusters, with different weights (one cluster must be the „true" principal one, with the largest weight, to make sense) (http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/cmeans.html).

We proceed now with the in/out-door temperature data and the window. Setting to a medium value the FCM "shape coefficient" (to have weighting function relatively sharp, but allowing overlapping tails) and the number of clusters to 3 (the best, but it is free to experiment with other settings) the results of the clustering of the in-door temperature measurements can be seen in the Fig. 78. We can see that the FCM parameter setting works nicely, the majority of data points belongs definitively to one or other cluster, data belonging to the transitions between the clusters is relatively sparse.

The FCM weighting function being non negative and normalized to 1 could in principle serve as a fuzzy membership function, but to build a fuzzy controller one usually prefers membership functions of a more standard behaviour like a Gaussian, a triangle, a trapeze. So the next step is to approximate the empirical clustered results with these standard shapes, see Fig. 78.

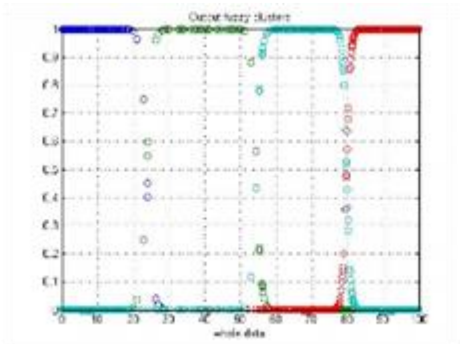


78. ábra. ab. Fuzzy clustering for the in-door temperature data.

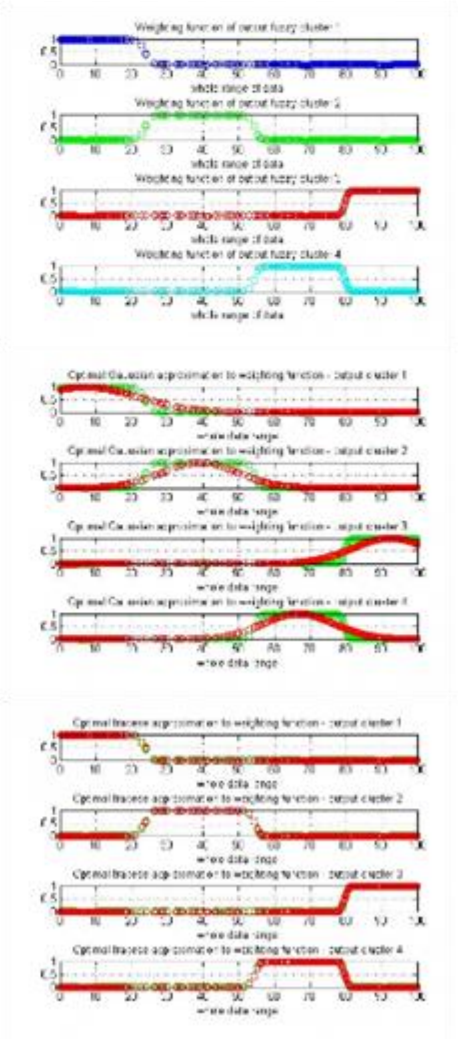


79. ábra. ab. Optimally approximating weights from Fig. ?? with Gaussian membership functions (left), and trapezoidal functions (right).

We can see that the trapeze approximation fares much better and this will be chosen for the controller. Similar results (3 clusters) can be obtained for the out-door temperature measurements. With the window opening the situation is slightly more complicated, because the human inhabitant used to shut the window, to open it fully, but then to open it slightly, or to open it almost fully. So here we work with 4 clusters, approximating them later with the trapezoidal membership functions, see Fig. 80.



80. ábra. Fuzzy clustering for the window-opening data.



81. ábra. abc. Optimally approximating weights from Fig. ?? with Gaussian membership functions (left), and trapezoidal functions (right).

When all of the involved universes are covered by the fuzzy membership functions, it is time to combine membership functions into fuzzy rules. In the simplified example we will seek rules in the form:

Fuzzy-set-Temperature-In-door AND Fuzzy-set-Temperature-Out-door

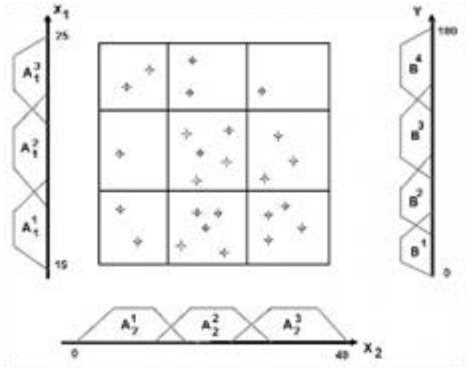
THEN Fuzzy-set-Window-opening

or (in the notation of Fig. 23): A_1^k AND A_2^n THEN B^m

In the Fig. 82. we can see that covering input universes with fuzzy sets creates a cell grid with the cells limited with 1/2 membership values. A particular measurement point at time point t:

(In-door Temp(t), Out-door Temp(t), Window-opening(t))

is represented in the grid by a dot belonging to one of the cells.



82. ábra. Cell-grid defined by the 1/2 value ordinates belonging to the input membership functions (left, bottom). For the reference also the output universe is depicted (right).

All dots (measurement data) belonging to the same cell share the fuzzy premise (e.g. in Fig. 83. data belonging to the cell marked in red share the premise A_1^2 AND A_2^2), but to various extend, dependent on their position within the cell. Data in the center, belonging to the higher values of the membership functions in the premise, is more representative, data dots at the borders, falling close to the 1/2 value of the membership functions, is considered less representative. This can be express by the weight of the data dot w(data) = A_1^2(data) x A_2^2 (data).

Tentatively let us create a rule form every data point in a form: $A_1^k(t)$ AND $A_2^n(t)$ THEN $y(t)$, where the premise membership functions are those which yield the maximum value for that particular data, and the output of the rule is for the time being the actual opening of the window.

The problem we face now is that although the data point falling into the same cell share the fuzzy functions in the premise, the measured window opening activity could be quite inconsistent. So looking for which output fuzzy membership function yields maximum to $y(t)$ necessarily will lead to different conclusions for the same premise, a situation forbidden in any logic. Another problem is that that way we have so many rules as the data points, and those were made numerous to fight the curse of dimensionality. The last step in the one-shot learning is to compress the number of rules into a manageable rule base.

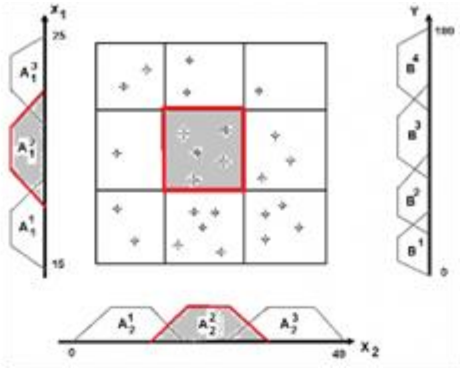
Clearly all the data (tentative) in a given cell share the rule premise (because the cell is defined by particular fuzzy sets from the input universes). It means that all these rules should be collapsed into one. But what would be the conclusion fuzzy set of such rule?

Accordingly to Wang-Mendel method we put into work the data point weights w(data) and compute an "average" conclusion of the cell as:

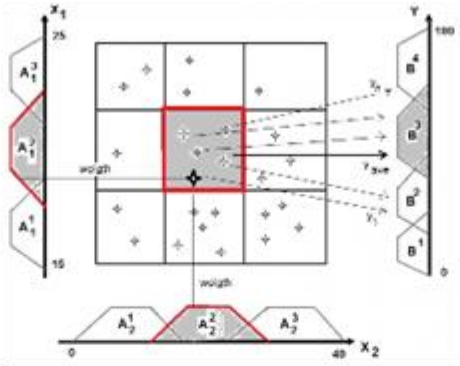
$$y_{average} = \Sigma\ y(t)w(t)/\Sigma w(t)$$

where t runs through all the data points belonging to the investigated cell. Last step is to check which output fuzzy set yields the highest membership value for the $y_{average}$ and pick it as the fuzzy consequence of the rule, see Fig. 84.

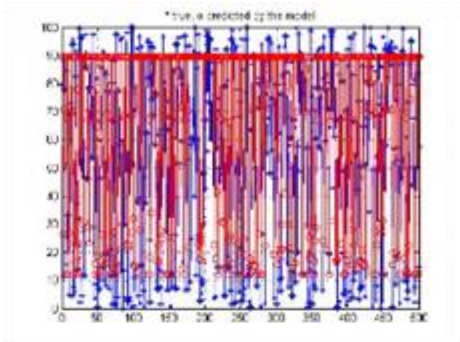
That way we will have at most so many rules as the number of cells (9 in our example), if every cell has enough data.



83. ábra. A particular cell in the input cell-grid, defined by particular input membership functions.

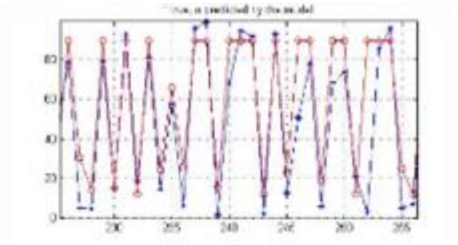


84. ábra. Computation of the conclusion fuzzy set of a rule.



85. ábra. Comparison of the system window-opening policy with the human behaviour consistent with the data used for learning system. The system performs well, what can be better visible on shorter time intervals, see e.g. Fig. ??.

The rules learned in the example are tested compared with the further behaviour of the room inhabitant consistent with his or her actual heating/ cooling policy. We can see that within an unavoidable (stochastic) margin the system adapted itself well to the human, see Fig. 85, and expanded fragment in Fig. 86.



86. ábra. A fragment of the system validation, from Fig. 14.26. We can see that the system is more step-like when it comes to almost fully opening the window (100%) what can be caused by the centroid mechanism to compute the output control variable.

The properly tuned system is working correctly until the human will change the preferred policy (which can happen due to the change in the external conditions, or due to change in mood or state of health, for example). The discrepancy between what system does and what is expected to do, will grow, until it becomes intolerable and the inhabitant will step in and redo the system actions. The system in such case is expected to accept user's actions, to memorize them, and if they become frequent, to use these new data points to:

1. recompute the output fuzzy sets in the rules,

2. even to recompute the clustering of the input fuzzy sets, accordingly to the one-shot learning methodology.

15.3. 14.3 Predicting and recognizing activities, detecting normal and abnormal behavior

In several intelligent embedded systems the detection and characterization of the activities of the person living or working in the environment is the most important task the system should perform. For example in an ambient assisted living (AAL) application helping an old, ill or handicapped person we are interested in the sleep periods, in the motion, in the toilet usage, in the nutrition of the person etc. Some of these are simple (e.g. sleeping), others are complex activities consisting of several actions.

The main problem is that the actions cannot be measured directly (there is no "breakfast detector"), we have two ways:

- Inferring the activity relying on some very simple detectors' measurements (motion, energy consumption, door contact ec.),

- Using surveillance cameras, in that case very hard image processing problems should be solved; ethical and human rights issues should be faced.

In this section the first approach is shown using the typical/abnormal human behavior detection scenario.

If characterization of the actions is needed, then the typical and abnormal actions should be identified. It is not a simple task to define what is typical, usually the regular; (semi)periodic events are taken as typical ones.

If the activity should be predicted some model of the person is needed. This model could be an a priori one, or could be learnt during the observation of the person.

15.3.1. 14.3.1 Recognizing activities

There are two basic ways in recognizing activities.

- A. If we have a model of the activity to be recognized, then inference based on this model and on noisy and shallow measurements should be performed. It should be mentioned that modeling an activity (especially a complex one) is a very hard problem. The model has several components, some of them may be missing, others are always present, the order of the elements may change etc.

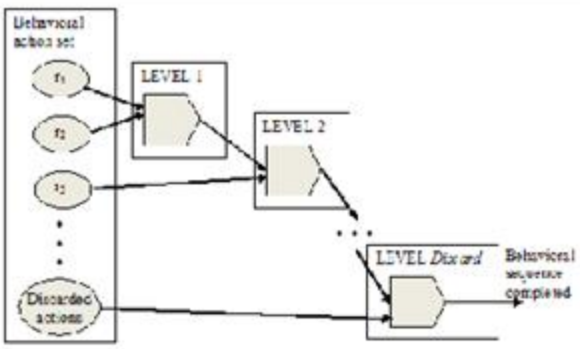
- B. If we have no exact model, the typical activities should be extracted from the observations.

15.3.2. 14.3.2 Model based activity recognition

There are several problems and suggestions for human activity modeling. A Finite Action-Set Learning Automaton is used to describe the human behavior. During the analysis the strengths and weaknesses of the model are emphasized as well. The finite-state assumption is a natural approach: there are several thousands of human actions, but the number of important ones for a special human or human group could be constrained to about some hundred ones.

In the model a human action is given by a triple $h_k =< l_k, s_k, t_k >$ where l_k is the label of the action (e.g. breakfast), t_k is a timestamp showing the time of occurrence, and the action should be uniquely determined by means of a sensor event s_k. This last assumption is a weakness of the model, in real life several actions cannot be identified based on one sensor event, only some probabilistic characterization of the occurrence is possible. (It is true, even if one sensor event is not a simple standalone sensor's finding, but a complex event determined by several sensors.)

The most important component of the model is the behavioral action sequence. It is given by a tuple $B^{(j)} =< H, R^{(j)}, C^{(j)} >$ where $H = \{h_1, h_2, h_3, ..., h_n\}$ is the set of possible human actions (or tasks), $R^{(j)} = \left\{h_1^{(j)}, h_2^{(j)}, h_3^{(j)}, ..., h_{|R^{(j)}|}^{(j)}\right\} \subseteq H$ is a subset of the available actions: these are required to complete the behavioral sequence. The most problematic part of such complex action-series is characterized by the $C^{(j)}$ set of temporal constraints among the actions of $R^{(j)}$. The basic tool is a partial order relationship operator: $r_A^{(i)} \prec r_B^{(i)} \Leftrightarrow t_A < t_B$. It means that action A precedes action B if the timestamp associated to A is lower (earlier) than that of associated to B. (This is a serious simplification of the time interval inference relations method suggested by Allen, see chapter 13.) The graphical representation of the behavioral action sequence is shown in figure 87. This shows the way of recognition as well: if the first action (first- defined by the temporal precedence constraints) is performed (recognized by the sensor event), then the second one is looked for, if it is detected as well, then the third one should occur etc. If all the actions of the sequence were performed in the proper order, then the behavioral sequence is completed. The order of tasks is represented by the levels associated to the task. It is important to note that levels are not describing hierarchy of actions, just the sequence of them. (In level 1 the proper sequence of the first 2 actions is detected, in level 2: after the first 2 actions the proper 3^{rd} on occurred etc).



87. ábra. Basic architecture for recognition of a behavioral action sequence

Of course this strict cascade behavior is too rigid even for simplified scenarios. In real life the order of some actions could be changed; some actions could be even missing from normal sequences. (For example during breakfast sometimes we drink first and only later eat some bread, in other cases we first eat something and only drink at the end. In some cases we only drink a cup of tea or coffee.) For that variability the model uses probability distribution for each level of each action. A p_{ij} probability is assigned to the i th event being performed on the j th level. In Table 11 probability distribution of a 4-action sequence is shown.

Action	$Level_1$	$Level_2$	$Level_3$	$Level_4$	Discard level
r_1	0.88	0.12	0	0	0
r_2	0.2	0.8	0	0	0
r_3	0	0.3	0.2	0	0.5
r_4	0.1	0.2	0.4	0.2	0.1

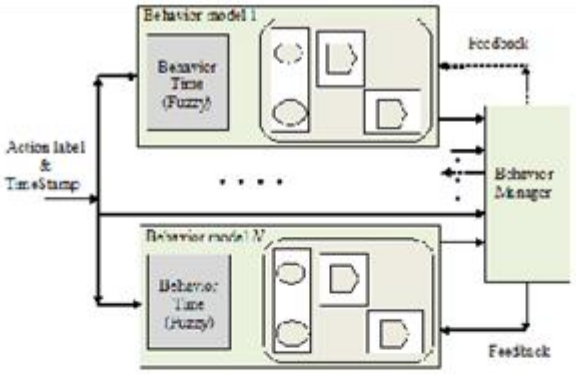
11. táblázat. Probability distribution of a 4-action behavioral sequence

The most probable sequence of the behavior modeled by table 11 is $< r_1 - r_2 - r_4 >$ (r_3 is discarded), but for example $< r_1 - r_2 - r_3 - r_4 >$ or $< r_2 - r_1 - r_3 - r_4 >$ are possible sequences as well.

The strength of this method is that the order of the actions is flexible; a weakness of it is that the typically hierarchical nature of the behavioral sequences is not modeled.

To detect a real-life behavioral sequence not only the actions involved, and the order of the actions are important, but the timing within the day as well. The behavior time could be used in several ways, e.g. Gaussian fuzzy membership functions are used to assign membership degree to every behavioral sequence. If the degree decreases below a given level the behavior is completed or aborted.

The system handling N parallel behavior sequences is shown in Figure 88.



88. ábra. System handling N parallel behavior sequences

The goals of the behavior manager are to activate the behavior models, to analyze and detect normal or abnormal behavior sequences, to abort the unfinished ones etc. It gives some feedback, and adapts the system to changing parameters.

The algorithm assumes that an off-line learning process was performed earlier, and a set of learned behaviors are used.

The algorithm consists of the following steps:

- a.) Initialization
- b.) Finding the active behavioral sequences
- c.) Checking of active behaviors; whether they are finished, aborted or abnormally finished
- d.) Processing of the current action
- e.) Updating the temporal fuzzy memberships of the current action
- f.) Checking violations of action order
- g.) Checking finished behaviors (normal, abnormal)
- h.) goto b.)

The most important weakness of the approach is that the model of the behavior sequence has to be built up; usually a rich structure is needed. Unfortunately these models are more or less individual for each person. (Some people have breakfast, others eat first at lunchtime, and some people go to bed in the same time, other people in irregular times etc.)

15.3.3. 14.3.3 Typical activities characterized by frequency and regularity

The other group of methods are model-free, no initial structure of the typical behavioral sequence are to be built. We will demonstrate this second approach. It shows the most important problems to be solved in that approach.

The first problem is that if we have no real model of the activity sequence, we somehow should define what is to be looked for. Because we are interested in typical activities, we assume that the activity is important for us, if it occurs several times, probably regularly. It could be periodic or at least quasi-periodic, but some activities are important even if they occur randomly. The basic theoretical assumption applied was to take the activity sequence; and search for episodes (activity subsequences), which could be used for a minimal description of the series.

According to Rissanen's Minimum Description Length Principle the episode, which gives the possibility of the shortest description is the most important one. This problem could be viewed as finding the length of the shortest program that outputs the data sequence of interest. It is called the Kolmogorov complexity of the data. Unfortunately there exists no algorithm that gives the shortest description for arbitrary sequences of data. Second problem is that Kolmogorov complexity depends on what computer language is used. The programming language could be chosen arbitrarily, but it has an influence on the complexity. Fortunately using any reasonable language and method the result will usually be acceptable, especially the importance of different episodes are comparable (the more important one gives possibility for a shorter description).

Example 14.1

Let us take the following series of 31 actions, each letter means a definite elementary action. The timestamp is a simple serial number in this demonstrative example.

A	B	B	A	B	B	A	B	B	A	D	E	B	B	A	...
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...
D	A	C	B	B	A	T	H	E	B	B	A	B	B	A	D
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

12. táblázat. Action sequence of 31 elements

If both one elementary action and one timestamp are encoded in length L, then the total length of the description is $2^*31^*L = 62^*L$, or $(1+31)^*L = 32^*L$ if we know that the series is equidistant. (In latter case only 1 start timestamp and all the actions are needed.) By using the frequent episode BBA, the following exact encoding of the action series could be used:

A	D	E	D	A	C	T	H	E	D				...
1 B	11 B	12 A	16	17	18	22	23	24	31				...

Table 14.3 Action sequence of Table 12 compressed using BBA important episode

The episode is cut from the sequence and in the first and second row the remaining elementary actions and their timings are shown, in the third row the episode for data compression is given. The original series is encoded this way using $(2^*10+3)^*L = 23^*L$. Using this episode a shorter encoding is possible: it means that BBA is probably an important part of the series. The compression ratio 23/32 characterizes the measure of how typical that episode is. If we would have another episode, which has a lower compression ratio, we assume it to be more typical.

(This example could be given using simple everyday activities as well. We can consider the actions a person takes each minute of each day of a year, but we can give a much more compact description stating that he/she eats 2 eggs and a piece of bread for breakfast each morning at 8:00. It compresses the original one year action stream. Another typical episode could be that the person gets his/her pension on the 15th day of every month. Obviously using this second episode the series could be compressed less than using the everyday one.)

There is no algorithm for finding the shortest possible description. Nevertheless reasonable algorithms could be found. One such algorithm consists of 2 steps.

A. Partition of the input sequence to find maximal episodes

The event series is incrementally processed using a moving window (so called event folding window). The goal is to find maximal episodes. Of course limits must be used to avoid meaningless long episodes: for example the whole series could be taken as one long episode (meaningless although it is really maximal). Two limits are suggested:

- the possible maximal number of events in the window (so called capacity of the window),
- the possible maximal time span between the first and the last action of the episode (time duration of the window).

Example 14.2

Let the event series be $<(a,1)(b,4)(a,9)(c,11)(a,17)(d,28)>$ each pair consist of an action (coded by a letter) and a timestamp. If the time window used has a capacity of Cw=3 and a time duration tw=12, then the following maximal episodes are found.

Action	Actions in window	Start time of the sliding window	Stop time of the sliding window	Maximal episodes just found
(a,1)	(a,1)	1	1	
	(a,1)	1	2	
....	....	....	....	....
(b,4)	<(a,1) (b,4)>	1	4	
	<(a,1) (b,4)>	1	5	
....	....	....	....	
(a,9)	<(a,1) (b,4) (a,9)>	1	9	<a b a>
....	....	....	....	....
(c,11)	<(a,1) (b,4) (a,9) (c,11)>	1	13	<b a c>
	<(b,4) (a,9) (c,11)>	2	14	
....	....	....	....	....
(a,17)	<(a,9) (c,11) (a,17)>	5	17	<a c a>
	<(a,9) (c,11) (a,17)>	6	18	
....	....	....	....	....
(d,28)	<(a,17) (d,28)>	16	28	<a d>
<end 40>				

Table 14.4 Maximal episodes of the $<(\mathrm{a},1)(\mathrm{b},4)(\mathrm{a},9)(\mathrm{c},11)(\mathrm{a},17)(\mathrm{d},28)>$ sequence using window capacity $C_w = 3$ and maximal episode (window) duration of $t_w = 12$.

B. Create candidates

All the maximal episodes found in step A are candidates. But it may happen that only a subset of the maximal episode is important, therefore all the possible subsets (the power set of the maximal episode) are possible candidates as well. (If in the previous 14.2 example $<\mathbf{b\ a\ c}>$ was a maximal episode then all $<\mathbf{b}>, <\mathbf{a}>, <\mathbf{c}>, <\mathbf{b\ a}>, <\mathbf{a\ c}>$ are possible important episodes as well.) Unfortunately taking into account the power sets of all maximal episodes found may produce intractable number of episodes.

To prune the candidate space not all the subsets are used, only if one of the following two criteria is met:

- The subset represents the intersection of two maximal episodes. (Because the intersection occurs in two episodes it may be more frequent than any of the parent episodes.)
- The subset represents the difference between a maximal episode and one of its episode subsets, which subset was found interesting.

These two rules are applied until all the possible candidates are found.

Example 14.3

Let us take three maximal episodes: $< a\ b\ c\ d >$ and $< a\ b\ c\ e >$ and $< a\ b\ e\ f >$. The candidates generated (and later evaluated using the compression potential) are shown in Table 14.5.

Maximal episode	List of generated candidates	Comment
<a b c d>	<a b c d>	
<a b c e>	<a b c d> , <a b c> , <a b c e>	Original max episodes and the intersection of them.
<a b e f>	<a b c d> , <a b c> , <a b c e> , <a b e f> , <a b> , <a b e>	New candidates: new max episode, new intersection subsets.
	???? : <d> , <e> , <c d> , <c e> , <e f> , , <e>, <f>	Difference subsets between the intersection subset and the original maximal episodes are taken only if the difference subset was found important.

Table 14.5 Important episode candidates generated from 3 maximal episodes.

The candidate generation shown in Table 14.5 is performed in 3 steps. Having the first maximal episode it is taken as a candidate. The second maximal episode is a candidate itself, but the intersection of the two episodes produces a third candidate.

C. Compute compression ratios

Compression ratios of all the generated candidates are evaluated; the candidates having the best compression ratios are selected.

As mentioned the calculation of the best compression ratio a candidate could provide is not trivial. Usually long periodic episodes with short repetition time give good compression possibilities. But everyday examples show that some episodes could have more complex pattern of timing: for example someone goes to the club every Tuesday and Friday. This means the timing pattern of the episode will be: 72, 96, 72, 96,...which could be used in a better compression than the 168 hours strict periodicity of two episodes. Another important question of the granularity: some events are periodic using day or hour resolution (e.g. breakfast, lunch), but they have a strong random nature if minute or second resolution is used.

In some cases totally aperiodic episodes could be important ones. In the MDL framework if an aperiodic episode is long, then it has a compression potential: only the start times of it should be stored as many times as it occurs, the long episode is to be stored only once.

Therefore the granularity problem and the proper way of compression should be solved at least a good approximations are needed. Nevertheless using some predefined structures (periodic, double-time pattern, aperiodic etc.) the best compression ratios assigned to the candidates could be evaluated and compared to each other. The candidates having the lowest compression ratios are taken as important ones.

It should be mentioned that emphasizing the typical events is not the only reasonable approach. In some cases (fault detection, diagnosis) the rare events are typically more important (heart attack is not a frequent event in life), but if we have information about the typical ones, it could help in finding the outliers (anomalies). Without knowing the typical the detection of atypical could be much harder.

15.3.4. Appendix A. References and suggested readings

15.3.4.1. Sensor fusion

- S. Challa and D. Koks, Bayesian and Dempster-Shafer fusion, Sadhana Vol. 29, Part 2, April 2004, pp. 145-174.
- M. Duta and M. Henry, The Fusion of Redundant SEVA Measurements, IEEE Trans. On Control Systems Technology, Vol. 13. No 2., March 2005, pp. 173-184.

- K. Faceli, A.C.P.L.F. De Carvalho and S.O. Rezende, Combining Intelligent Techniques for Sensor Fusion, Applied Intelligence Vol. 20. 2004. pp 199-213
- V. Gupta and N. C. Martins, On Fusion of Information from Multiple Sensors in the Presence of Analog Erasure Links, Proc. Of the Joint 48^{th} IEEE Conf. on Decision and Control and 28^{th} Chinese Control Conference, Shanghai, P.R.China, Dec 16-18. 2009., pp 2705-2710.
- D.L. Hall and J. Llinas, An Introduction to Multisensor Data Fusion, Proceedings of the IEEE Vol. 85. No 1. Jan. 1997. pp 6-23
- M. Kalandros, L. Trailovic, L.Y. Pao and Y. Bar-Shalom, Tutorial on Multisensor Management and Fusion Algorithms for Target Tracking, Proc. Of the 2004 American Control Conference, Boston Massachusetts, USA June30-July 2, 2004, pp. 4734-4748
- N.S.V. Rao, D.B. Reister and J. Barhen, Information Fusion Methods Based on Physical Laws, IEEE Trans. On Pattern Analysis and Machine Intelligence, Vol. 27. No 1., Jan 2005, pp. 66-77.
- T. Wimalajeewa and S. K. Jayaweera, Power Efficient Analog Forwarding for Correlated Data Fusion in Wireless Sensor Networks, Proc of the 66th IEEE Vehicular Technology Conference (VTC'07 - Fall) Baltimore, MD, USA, Oct. 2007, pp. 367-371.
- H. Wu, M. Siegel, R. Stiefelhagen and J. Yang, Sensor Fusion Using Dempster-Shafer Theory, Proc. Of the IEEE Instr. and Measurement Technology Conf., Anchorage, AK, USA, 21-23 May 2002, pp. 7-12.
- H. Wu, M. Siegel and S. Ablay, Sensor Fusion Using Dempster-Shafer Theory II: Static Weighting and Kalman Filter-like Dynamic Weighting, Proc. Of the IEEE Instrumentation and Measurement Technology Conference, Vail, CO, USA, 20-22 May 2003, pp. 907-912.

15.3.4.2. Ambient intelligence, intellligent spaces

- The No.1 web site for Ambient Intelligence ...http://www.ambientintelligence.org/
- Ambient Intelligence/ electronic book http://www.emergingcommunication.com/volume6.html
- Fluid Interfaces Group (formerly Ambient Intelligence Group, MIT Media Lab) http://ambient.media.mit.edu/
- Smart Houses, Rooms and Appliances, AAAI Topics http://www.aaai.org/AITopics/pmwiki/pmwiki.php/AITopics/SmartHouses
- Seminar: Intelligent Spaces, TU Dortmund http://www.irf.tu-dortmund.de/cms/en/IS/Teaching/SS07/IntelligentSpaces/index.html
- Ambient Intelligence Lab, CMU http://www.cmu.edu/vis/projects.html
- Siemens, Rest and Nursing Homeshttp://www.buildingtechnologies.siemens.com/BT/GLOBAL/EN/PRODUCTS_SYSTEMS/ET/GAMMA/SOLUTIONS/Pages/solutions.aspx
- J. of Ambient Intell. and Smart Environments, http://www.iospress.nl/loadtop/load.php?isbn=18761364
- PEIS Ecology - Ecology of Physically Embedded Intelligent Systems, http://aass.oru.se/ $\sim$ peis/frameset_page.html
- Aware Home http://awarehome.imtc.gatech.edu/
- European Research Consortium for Informatics and Mathematics www.ercim.org
- Special Issue on Ambient Intelligence, ERCIM NEWS, Nr 47 Oct 2001,http://www.ercim.org/publication/Ercim_News/enw47/EN47.pdf
- OZONE: New Technologies and Services for Emerging Nomadic Societies,http://www.hitech-projects.com/euprojects/ozone/

- EUNICA: iExtensible Universal control of Appliances intelligent household system,http://www.ercim.org/publication/Ercim_News/enw47/bielikova.html
- inHaus: innovation Center for the Intelligent House, Duisburg,http://www.eurescom.de/message/messageMar2003/inHaus_Duisburg.asp
- Living Room: Tampere University of Technology,http://www.ele.tut.fi/research/personalelectronics/projects/smart_home.htmhttp://www.ele.tut.fi/research/personalelectronics/projects/ekoti_03/index.htm
- Adaptive House, ACHE - Adaptive Control of Home Environmentshttp://www.cs.colorado.edu/ ~ mozer/house/
- Lessons from an Adaptive Home,http://www.cs.colorado.edu/% 7Emozer/papers/reprints/smart_environments.pdf
- An intelligent environment must be adaptivehttp://www.cs.colorado.edu/% 7Emozer/papers/ieee.html
- The neural network house: An environment that adapts to its inhabitants,http://www.cs.colorado.edu/% 7Emozer/papers/nnhadapt.html
- The Neurothermostat: Predictive optimal control of residential heating systems,http://www.cs.colorado.edu/% 7Emozer/papers/neurothermostat.html
- Intelligent Home http://intelligenthomes.com/
- House_n: http://architecture.mit.edu/house_n/
- The PlaceLab http://architecture.mit.edu/house_n/placelab.html
- IBM Context Sphere environment for developing and executing context-sensitive applications,http://www.research.ibm.com/cxs/index.html
- CALO: Cognitive Assistant that Learns and Organizeshttp://caloproject.sri.com/
- COGAIN: network of excellence on Communication by Gaze Interaction http://www.cogain.org/
- Sougdo City: A Master Plan Inspired By The World http://www.songdo.com/default.aspx
- U-City, Korea Leading Global u-City http://www.koreaittimes.com/story/4371/leading-global-u-city
- ALADIN - Ambient Lighting Assistance for an Ageing Populationhttp://www.ambient-lighting.eu/files/aladin_presentation.pdf
- MavHome http://ailab.eecs.wsu.edu/mavhome/index.html
- The Gator Tech Smart House: A Programmable Pervasive Space, Computer March 2005,http://www.icta.ufl.edu/projects/publications/helal_GTSH_IEEE_Computer_March_2005.pdf
- iDorm http://cswww.essex.ac.uk/Research/iieg/idorm.htm
- iDorm Ambient Intelligent Agent Demonstration, http://www.youtube.com/watch?v=NtxkEYLeLZc
- Siemens Smart Home, http://www.buildingtechnologies.siemens.com/bt/global/en/Home/Pages/home.aspx
- Philips http://www.philips.hu/#/headernav/consumer/
- Smart Kindergarten http ://nesl.ee.ucla.edu/projects/smartkg/
- KidsRoom http://vismod.media.mit.edu/vismod/demos/kidsroom/kidsroom.html
- Web sites for other smart homes and smart technology, http://www.smarthome.duke.edu/program/urls.php

- RoboCare: to build a multi-agent system which generates user services for human assistance, http://robocare.istc.cnr.it/
- Taxonomy of Smart Houses http://www2.imm.dtu.dk/∼cdj/SmartHouseWebSite/taxonomy.html
- Intelligent Houses Animated Scenarios http://www2.imm.dtu.dk/∼cdj/SmartHouseWebSite/scenarios.html
- The Intelligent Room, MIT http://www.ai.mit.edu/ARCHIVE/old-projects/abstracts2000/pdf/z-mhcoen1.pdf
- Biosphere 2 http://en.wikipedia.org/wiki/Biosphere_2, http://www.b2science.org/,http://www.biospherics.org/phototour.html
- BIOS-3 http://en.wikipedia.org/wiki/BIOS-3
- Ambient Intelligence-the Next Step for Artificial Intelligence
- Analysis of AmI Scenarios. Safeguards in a World of Ambient Intelligence (SWAMI)
- Scenarios for Ambient Intellligence in 2010, ISTAG

15.3.4.3. Ambient Assisted Living (AAL) - Ambient Assisted Cognition (AAC) - Activities of Daily Living (ADL)

- A New Research Challenge: Persuasive Technology to Motivate Healthy Aging, http://web.media.mit.edu/∼intille/teaching/fall04/papers/TITB.pdf
- The Independent LifeStyle Assistant (I.L.S.A.): Deployment Lessons Learned,http://www.cs.cmu.edu/∼khaigh/ILSAEXTERNALWEBSITE/content/publications/04-Haigh-AAAI-WS.pdf
- The Independent LifeStyle Assistant (I.L.S.A.): AI Lessons Learned,http://www.cs.cmu.edu/∼khaigh/papers/haigh-iaai04.pdf
- User Modelling in Ambient Intelligence for Elderly and Disabled People,http://www.glyndwr.ac.uk/groutv/papers/umaiedp.pdf
- An Ambient Assisted Living System for Telemedicine with Detection of Symptoms,http://link.springer.com/chapter/10.1007% 2F978-3-642-02267-8_9
- Activity Recognition for Context-aware Hospital Applications: Issues and Opportunities for the Deployment of Pervasive Networks
- The Independent LifeStyle AssistantTM (I.L.S.A.): Lessons Learned,http://www.cs.cmu.edu/afs/cs/user/khaigh/www/ILSAEXTERNALWEBSITE/content/publications/ilsa-honeywell-techreport.pdf
- Living Assistance Systems - An Ambient Intelligence Approach,http://www.irisa.fr/lande/lande/icse-proceedings/icse/p43.pdf
- HYCARE: A Hybrid Context-Aware Reminding Framework for Elders with Mild Dementia,http://media.cs.tsinghua.edu.cn/∼qinwj/publications/icost08.pdf
- GerAmi: Improving Healthcare Delivery in Geriatric Residences
- The use of brain-computer interfacing for ambient intelligence
- Ambient Interfaces for Elderly People at Home
- A Living Lab for Ambient Assisted Living in the Municipality of Schwechat
- Audiovisual Sensing of Human Movements for Home-Care and Security in a Smart Environment
- Social Rhythms and Nocturnal Routines in Community Dwelling Older Adults

15.3.4.4. Agents in Ambient Intelligent Applications

- A Component-Based Ambient Agent Model for Assessment of Driving Behaviour
- A Generic Topology for Ambient Intelligence
- A Multi-agent Approach to Controlling a Smart Environment (MavHome project)
- A Multi-dimensional Model for Task Representation and Allocation in Intelligent Environments
- Adaptive Estimation of Emotion Generation for an Ambient Agent Model
- Adding Intelligence to Ubiquitous Computing Environments (iDorm project)
- Agent Patterns for Ambient Intelligence
- Mobile Agents for Ambient Intelligence
- Agents Visualization in Intelligent Environments
- Building Brains for Rooms: Designing Distributed Software Agents (Scatterbrain project)
- Causal Reasoning for Alert Generation in Smart Homes
- Creating an Ambient-Intelligence Environment Using Embedded Agents (iDorm project),http://cswww.essex.ac.uk/staff/vic/papers/2004_IS04%28CreatingAmbientIntelligence%29SXversion.pdf
- Design Principles for Intelligent Environments
- Inhabited intelligent environments (iDorm, iDorm-2 project)
- Interface agents: A review of the field
- Learning Temporal Relations in Smart Home Data (MavHome project)
- Rascal - a Resource Manager For Multi Agent Systems In Smart Spaces
- Realtime Role Coordination For Ambient Intelligence (PEACH - active museum projekt)
- Software Agents for Ambient Intelligence
- Spatiotemporal Reasoning for Smart Homes
- Spatio-Temporal Reasoning and Context Awareness
- Spatial and Temporal Reasoning for Ambient Intelligence Systems
- Temporal Constraints with Multiple Granularities in Smart Homes
- Using Interval-Based Reasoning in Smart Homes
- Using Event Calculus for Behaviour Reasoning and Assistance in a Smart Home
- Keeping the Resident in the Loop: Adapting the Smart Home to the User

15.3.4.5. Fuzzy learning and control in Intelligent Spaces

- Using FML and Fuzzy Technology in Adaptive Ambient Intelligence Environments
- Monitoring the State of a Ubiquitous Robotic System: A Fuzzy Logic Approach

- Design Of Fuzzy Controllers
- A Fuzzy Incremental Synchronous Learning Technique for Embedded-Agents Learning and Control in Intelligent Inhabited Environments (iDorm)
- Proactive Fuzzy Control and Adaptation Methods for Smart Homes
- L. X.Wang and J. M. Mendel, Generating fuzzy rules by learning from examples, IEEE Trans. Syst., Man, Cybern. , vol. 22, pp. 1414-1427, Dec 1992.
- L.X. Wang, The WM Method Completed: A Flexible Fuzzy System Approach to Data Mining, IEEE Trans. on Fuzzy Systems , Vol. 11, No. 6, Dec 2003, pp. 768-782,
- A Third-Generation Telecare System using Fuzzy Ambient Intelligence
- A Fuzzy Based Architecture for Learning Relevant Embedded Agents Associations
- F. Doctor, H. Hagras, and V. Callaghan, A Fuzzy Embedded Agent-Based Approach for Realizing Ambient Intelligence in Intelligent Inhabited Environments, IEEE Trans. on Syst., Man, Cybern. Part A: Systems and Humans, Vol. 35, No. 1, Jan 2005, pp. 55-65,
- Generating Fuzzy Rules by Learning from Examples
- Using Fuzzy Logic to Monitor the State of an Ubiquitous Robotic System, http://aass.oru.se/~peis/Papers/jus08.pdf
- Fuzzy C-means clustering, wiki
- Fuzzy C-Means, http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/cmeans.html

15.3.4.6. Timeseries Mining

- E. Keogh, J. Lin, A. Fu, HOT SAX: Efficiently Finding the Most Unusual Time Series Subsequence, Fifth IEEE Int. Conf. on Data Mining, 27-30 Nov. 2005, Houston, http://www.cse.cuhk.edu.hk/~adafu/Pub/icdm05time.pdf
- Experiencing SAX: a Novel Symbolic Representation of Time Series
- HOT SAX: Finding the Most Unusual Time Series Subsequence: Algorithms and Applications
- Detecting Time Series Motifs Under Uniform Scaling
- A Novel Bit Level Time Series Representation with Implications for Similarity Search and Clustering
- Lin, J., Keogh, E., Lonardi, S. and Chiu, B., A Symbolic Representation of Time Series, with Implications for Streaming Algorithms, 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, San Diego, June 13, 2003, http://homepages.abdn.ac.uk/yaji.sripada/pages/teaching/CS4031/information/SAX.pdf
- Visualizing and Discovering Non-Trivial Patterns In Large Time Series Databases
- B. Chiu, E. Keogh, S. Lonardi, Probabilistic discovery of time series motifs, 9th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, KDD'03, 2003, pp. 493-498,http://www.cc.gatech.edu/~isbell/reading/papers/p493-chiu.pdf
- SAX home page, http://www.cs.ucr.edu/~eamonn/SAX.htm
- Segmenting Time Series: A Survey and Novel Approach

15.3.4.7. Context computing and context management

- A Location Model for Ambient Intelligence
- Context Driven Observation of Human Activity
- Explanations and Context in Ambient Intelligent Systems
- Location Aware Resource Management in Smart Homes
- Semantic-level Understanding of Human Actions and Interactions using Event Hierarchy
- Towards a Better Understanding of Context and Context-Awareness
- Towards an extensible context ontology for Ambient Intelligence
- Cases, Context, and Comfort: Opportunities for Case-Based Reasoning in Smart Homes,http://www.springerlink.com/content/j25n788867026842/
- Towards a Better Understanding of Context and Context-Awareness, http://smartech.gatech.edu/jspui/bitstream/1853/3389/1/99-22.pdf
- Context Driven Observation of Human Activity, http://www.springerlink.com/content/8pwrf03hajarmwl6/
- http://homepages.inf.ed.ac.uk/rbf/CAVIAR/PAPERS/Crowley-Context-paper.pdf
- An Ontology for Context-Aware Pervasive Computing Environments, http://ebiquity.umbc.edu/_file_directory_/papers/64.pdf
- HYCARE: A Hybrid Context-Aware Reminding Framework for Elders with Mild Dementia, http://media.cs.tsinghua.edu.cn/qinwj/publications/icost08.pdf
- SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications, http://ebiquity.umbc.edu/_file_directory_/papers/105.pdf

15.3.4.8. Natural language interfaces

- A Reliable Natural Language Interface to Household Appliances, IUI f03, January 12-15, 2003, Miami, Florida, USA.,http://www.cis.temple.edu/∼ yates/papers/nlia-iui03.pdf
- Human-Computer Interaction: Overview on State of the Art, Int. J. on Smart Sensing and Intelligent Systems, Vol. 1, No. 1, March 2008, http://www.s2is.org/Issues/v1/n1/papers/abstract9.pdf
- A Reliable Natural Language Interface To Household Appliances - menu2dialog, 2nd IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems, Aug 5, 2001, Seattle,http://www.cis.temple.edu/∼ yates/papers/nlia-iui03.pdf
- Reasoning in Attempto Controlled English, Technical Report ifi-2002.01,http://attempto.ifi.uzh.ch/site/pubs/papers/TR.ifi-2002.01.pdf
- Towards a Theory of Natural Language Interfaces to Databases, IUI f03 Miami, Florida USA,http://homes.cs.washington.edu/∼ etzioni/papers/nli-iui03.pdf
- UIML: An Appliance-Independent XML User Interface Language,http://www8.org/w8-papers/5b-hypertext-media/uiml/uiml.html

15.3.4.9. Sensor Web

- NASA Sensor Web Technology for Autonomous Asset Tasking and Customized Data Delivery, NASASensorweb
- White Paper, Creation of Specific SensorML Process Models, Jan 27th, 2006 (Draft), Earth System Science Cent. - NSSTC, Univ. of Alabama (UAH)

- OGC Sensor Web Enablement: Overview and High Level Architecture , S. Nittel, A. Labrinidis, and A. Stefanidis (Eds.): GSN 2006, LNCS 4540, pp. 175-190, 2008. http://portal.opengeospatial.org/files/?artifact_id=25562
- G. Percivall, C. Reed, OGC® Sensor Web Enablement Standards, Sensors and Transducers Journal, Vol.71, Issue 9, Sept 2006, pp.698-706, OGC Sensor Web Enablement Standards
- Tutorial I: Using SensorML to describe a Complete Weather Station, Feb 27th, 2006, Earth System Science Cent. - NSSTC, Univ. of Alabama (UAH)
- Sensor Model Language (SensorML): XML-Based Language for In-situ and Remote Sensors, NIST Workshop on Data Exchange Standards at the Jobsite, May 29, 2003
- The German Indonesian Tsunami Early-Warning System (GITEWS), http://www.gitews.org
- http://sensorwebs.jpl.nasa.gov/
- S. Chien, B. Cichy, A. Davies, D. Tran, G. Rabideau, R. Castano, R. Sherwood, D. Mandl, S. Frye, S. Shulman, J. Jones, S. Grosvenor, An Autonomous Earth-Observing Sensorweb, IEEE Intelligent Systems, May/June 2005., http://sensorweb.nasa.gov/new/validationReport/IEEE-SMC-2005-Sensorweb-v04.pdf
- Sensor Web URL list, Sensor Web Instructions, http://eo1.gsfc.nasa.gov/new/sensorwebexp/
- A. Bröring, J. Echterhoff, S. J., I. Simonis, T. Everding, C. Stasch, S. Liang and R. Lemmens, New Generation Sensor Web Enablement, Sensors 2011, 11, pp .2652-2699
- K. A. Delin, The Sensor Web: A Distributed, Wireless Monitoring System, Sensors, Apr 1, 2004.
- http://www.earthobservations.org/documents/sbas/di/42_the_german_indonesian_tsunami_ews.pdf
- http://sensorwebs.jpl.nasa.gov/
- http://eo1.gsfc.nasa.gov/new/sensorwebexp/ Sensor Web URL list, Sensor Web Instructions

15.3.4.10. Smart Houses

- Perspectives of ambient intelligence in the home environment, Telematics and Informatics 22 (2005) 221-238Designing a Home of the Future, Pervasive computing, Volume 1 Issue 2, April 2002, Page 76-82
- How Smart are our Environments? An Updated Look at the State of the Art, Journal of Pervasive and Mobile Computing, Volume 3 Issue 2, March, 2007, Pages 53-73
- Temporal Pattern Discovery for Anomaly Detection in a Smart Home, Proc. of the 3rd IET Int. Conf. on Intell. Environments (IE 07), Germany, Sept 2007
- Mining Sensor Data in Smart Environment for Temporal Activity Prediction, ACM KDD[81?]f07, August 12-15, 2007, San Jose, California, USA
- Smart Homes Can Be Smarter, J.C. Augusto and C.D. Nugent (Eds.): Designing Smart Homes, LNAI 4008, pp. 1-15, 2006.
- The Role of Prediction Algorithms in the MavHome Smart Home Architecture, IEEE Wireless Communications, Dec 2002

15.3.4.11. Sensors - Devices - Human Computer Interfaces

- Accessing Ambient Intelligence through Devices with Low Computational and Communication Power, Adaptive Interfaces for Supportive Ambient Intelligence Environments, K. Miesenberger et al. (Eds.): ICCHP 2008, LNCS 5105, pp. 30-37, 2008.

- Ambient Intelligence and Multimodality, C. Stephanidis (Ed.): Universal Access in HCI, Part II, HCII 2007, LNCS 4555, pp. 33-42, 2007.
- Ambient Intelligence: Towards Smart Appliance Ensembles, M. Hemmje et al. (Eds.): E.J. Neuhold Festschrift, LNCS 3379, pp. 261-270, 2005.
- Bayesian Reasoning for Sensor Group-Queries and Diagnosis, R. Kotagiri et al. (Eds.): DASFAA 2007, LNCS 4443, pp. 522-538, 2007
- End User Tools for Ambient Intelligence Environments: An Overview, Human-Computer Interaction, Part II, HCII 2007, LNCS 4551, pp. 864-872, 2007.
- Engaging Personas and Narrative Scenarios, Lene Nielsen, PhD Dissertation, 2004, copenhagen business school, http://openarchive.cbs.dk/bitstream/handle/10398/6448/working% 20paper% 202004% 2016.pdf?sequence=1
- V. Prasanna et al. (Eds.): Organization in Sensor Networks, DCOSS 2005, LNCS 3560, pp. 191-205, 2005.
- Gaze as a Supplementary Modality for Interacting with Ambient Intelligence Environments, C. Stephanidis (Ed.): Universal Access in HCI, Part II, HCII 2007, LNCS 4555, pp. 848-857, 2007.
- Human-Based Sensing - Sensor Systems to Complement Human Perception, Int. J. on Smart Sensing and Intell. Systems, Vol. 1, No. 1, March 2008
- Human-Computer Interaction: Overview on State of the Art, Int. J. on Smart Sensing and Intell. Systems, Vol. 1, No. 1, March 2008
- Living with Hyper-reality, Y. Cai and J. Abascal (Eds.): Ambient Intelligence in Everyday Life, LNAI 3864, pp. 130 - 141, 2006.
- Multimodal Interfaces That Process What Comes Naturally, Comm. of the ACM March 2000/Vol. 43, No. 3
- Self-Organizing Sensor Networks, M. Bubak et al. (Eds.): ICCS 2004, LNCS 3038, pp. 1233-1240, 2004.
- Sensor networks for continuous health monitoring, BT Technology Journal, Vol 22, No 3, July 2004
- Sensor networks: an overview, IEEE Potentials April/May 2003
- Internet of Things, http://www.internet-of-things-research.eu/about_iot.htm
- FBI Biometric Center of Excellence,http://www.biometriccoe.gov/http://www.biometriccoe.gov/http://www.biometriccoe.gov/
- D. A. Norman and S. W. Draper, User Centered System Design: New Perspectives on Human-Computer Interaction, NJ: Lawrence Erlbaum Associates, 1986.
- F. Karray, M. Alemzadeh, J. Abou Saleh and M. N. Arab, Human-Computer Interaction: Overview on State of the Art, Int. J. on Smart Sensing and Intelligent Systems, Vol. 1, No. 1, March 2008
- https://en.wikipedia.org/wiki/Seven_stages_of_action

15.3.4.12. Planning

- Problems with Intent Recognition for Elder Care
- Logic Programming, Abduction and Probability, a top-down anytime algorithm for estimating prior and posterior probabilities
- Partial Observability and Probabilistic Plan/Goal Recognition
- Plan Recognition in Intrusion Detection Systems

- Plan-Based Configuration of an Ecology of Robots
- Probabilistic Horn abduction and Bayesian networks
- Probabilistic Plan Recognition for Hostile Agents
- Recognizing Plan/Goal Abandonment
- Representing diagnostic knowledge for probabilistic Horn abduction
- Formal theory of plan recognition and its implementation

15.3.4.13. Spatial and Temporal Inference

- P. van Beek, Approximation Algorithms for Temporal Reasoning, Proc. of 11th Int. Joint Conf. on Artificial Intelligence, Detroit, Michigan USA, pp 1291-96
- S.M.R. Dehak, I. Bloch, H. Maitre Spatial Reasoning with Incomplete Information on Relative Positioning, IEEE Trans. on Pattern Analysis and Mach. Intelligence, Vol. 27. No. 9. Sept. 2005, pp 1473-84
- D. Lymberopoulos, A. Bamis, A. Savvides Extracting Spatiotemporal Human Activity Patterns In Assisted Living Using A Home Sensor Network, PETRA'08, July 15-19, 2008 Athens, Greece

15.3.4.14. Intelligent Sensor Networks

- K. R. Fowler, The Future of Sensors and Sensor Networks, Survey Results Projecting the Next 5 Years, Proc. of Sensors App. Symp. 2009. SAS 2009. IEEE Sensors App. Symp. 2009. SAS 2009. IEEE, New Orleans, LA, USA, 17-19 Feb. 2009, pp. 1 - 6
- P. Gajbhiye and A. Mahajan, A Survey of Architecture and Node Deployment in Wireless Sensor Network, Proc. of Int. Conf. on App. of Digital Information and Web Technologies, 2008. ICADIWT, Ostrava, Czech Republic, August 4 - 6 2008, pp 426-430.
- P. Györke and B. Pataki, Energy aware measurement scheduling in WSNs used in AAL applications, Accepted to IEEE Trans. on Instrumentation and Measurement, 2013.
- V. Jelicic, T. Razov, D. Oletic, M. Kuri and V. Bilas, MasliNET: A Wireless Sensor Network based Environmental Monitoring System, Proc. on Information and Comm. Technology, Electronics and Microelectronics, MIPRO 2011, 23-27 May 2011, Opatija Croatia, pp. 150 - 155.
- R.V. Kulkarni, A. Förster, and G.K. Venayagamoorthy, , Computational Intelligence in Wireless Sensor Networks: A Survey, Comm. Surveys and Tutorials, IEEE , vol.13, no.1, pp.68-96, First Quarter 2011
- W. Lin, D. Li, Y. Tan, J. Chen and T. Sun, Architecture of Underwater Acoustic Sensor Networks: A Survey, Proc. of First Int. Conf. on Intell. Networks and Intell. Systems, 2008
- Ch. Lin, Y. He, Ch. Peng and L. T. Yang, A Distributed Efficient Architecture for Wireless Sensor Networks, Proc. of 21st International Conference on Advanced Information Networking and Applications Workshops, 2007, AINAW '07, Niagara Falls, Ontario, Canada 21-23 May 2007, pp 429-434
- G.H. Raghunandan and B.N. Lakshmi, A Comparative Analysis of Routing Techniques for Wireless Sensor Networks, Proc. of the Nat. Conf. on Innovations in Emerging Technology-201, Perundurai, Erode, Tamilnadu, India.17-18. February, 2011.pp.17-22.

15.3.4.15. Behavior modelling

- D. J. Ward and D. J. C. MacKay. Fast hands-free writing by gaze direction. Nature, 418:838, 2002.
- Dwight W. Read. How Culture Makes Us Human. Left Coast Press, Walnut Creek, CA, 2012.

- Susan S. Jones. Imitation and empathy in infancy. Cognition, Brain, Behavior, 13:391-413, 2009.
- Paul Ekman and Wallace F. Friesen. Unmasking the Face. Malor Books, Los Altos, CA, 2003.
- A. A. Marsh, H. A. Elfenbein, and N. Ambady. Nonverbal eaccents f: cultural differences in facial expressions of emotion. Psychology Science, 14:373-376, 2003.
- A. A. Marsh, H. A. Elfenbein, and N. Ambady. Separated by a common language: Nonverbal accents and cultural stereotypes about Americans and Australians. Journal of Cross Cultural Psychology, 38:284-301, 2007.
- H. Hill and A. Johnston. Categorizing sex and identity from the biological motion of faces. CurrentBiology, 11:880-885, 2001.
- Rachel Adelson. Detecting deception. Monitor on Psychology, 37:70, 2004.
- Mihaly Csikszentmihalyi. Finding Flow: The Psychology of Engagement With Everyday Life. Basic Books, New York, NY, 1998.
- Moataz El Ayadi, Mohamed S. Kamel, and Fakhri Karray. Survey on speech emotion recognition: Features, classification schemes, and databases. Pattern Recognition, 44:572-587, 2011.
- Shashidhar G. Koolagudi and K. Sreenivasa Rao. Emotion recognition from speech: a review. International Journal of Speech Technology, 15:99-117, 2012.
- Anton Batliner, Björn Schuller, Dino Seppi, Stefan Steidl, Laurence Devillers, Laurence Vidrascu, Thurid Vogt, Vered Aharonson, and Noam Amir. The automatic recognition of emotions by speech. Speech Communication, 53: 1062-1087, 2011.
- J. M. Saragih, S. Lucey, and J. F. Cohn. Deformable model fitting by regularized landmark mean-shift. Int. J. of Comp. Vision, 91(2):200-215, 2011.
- T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. IEEE Trans. on Pattern Analysis and Machine Intelligence, 23(6):681-685, 2001.
- C. M. Yang, Z. S. Lin, C. L. Hu, Y. S. Chen, L. Y. Ke, and Y. R. Chen. A novel dynamic sensing of wearable digital textile sensor with body motion analysis. In Proc. of the IEEE Conf. on Eng. Med. Biol. Soc., pages 4898-4901, 2010.
- N. N. Y. Chu, Chang-Ming Yang, and Chih-Chung Wu. Game interface using digital textile sensors, accelerometer and gyroscope. IEEE Trans. on Consumer Electronics, 58:184-189, 2012.
- I. Szita, B. Takács, and A. L"orincz. Epsilon-MDPs: Learning in varying environments. Journal of Machine Learning Research, 3:145-174, 2003.
- A. Lorincz, I. Pólik, and I. Szita. Event-learning and robust policy heuristics. Cognitive Systems Research, 4:319-337, 2003.
- I. Szita and A. Lorincz. Optimistic initialization and greediness lead to polynomial time learning in factored MDPs. In Léon Bottou and Michael Littman, editors, Int. Conf. on Machine Learning (ICML 2009), pp. 1001-1008, Montreal, June 2009. Omnipress.
- G. Hévízi, M. Biczó, B. Póczos, Z. Szabó, B. Takács, and A. L"orincz. Hidden Markov model finds behavioral patterns of users working with a head mouse driven writing tool. In Int. Joint Conf. on Neural Networks 2004, 2004. Paper No. 1268.
- I. Szita and A. Lorincz. The many faces of optimism: a unifying approach. In Andrew McCallum and Sam Roweis, editors, Int. Conf. on Machine Learning (ICML 2008), pp. 1048-1055, Helsinki, Finland, 2008. Omnipress.
- A. Lorincz and D. Takács. Agi architecture measures human parameters and optimizes human performance. In J. Schmidhuber, K.R. Thórisson, and M. Looks, editors, Conf. on Artificial General Intelligence (AGI-

2011), vol 6830 of Lecture Notes in in Artificial Intelligence, pp. 321-326, Berlin Heidelberg, 2011. Springer-Verlag.

15.3.4.16. Activity

- E.O. Heierman, G.M. Youngblood, D.J. Cook, Mining temporal sequences to discover interesting patterns, KDD'04, Workshop on Mining Temporal and Sequential Data, August 22-25, 2004 Seattle
- J. Rissanen, Stochastic Complexity in Statistical Inquiry, Word Scientific Publishing Company, 1989
- Online recognition of human activities and adaptation to habit changes by means of learning automata and fuzzy temporal windows, Information Sciences, Vol 220, January 2013, pp 86-101

15.3.4.17. HCI

- KurzweilAI, Lifeboat, TED, Scholarpedia, and Wikipedia. Up-to-date readings. Internet, 201x.http://www.kurzweilai.net/, www.ted.com, http://www.scholarped.org, http://www.wikipedia.org/
- EEG gadgets. Google image search. Search images with phrase "eeg electrodes".
- Mindwalker. EU project site. https://www.mindwalker-project.eu/
- Steve Dent. FDA clears Argus II 'bionic eye' for sale in the US. Engadget, 2013.http://www.engadget.com/2013/02/15/fda-clears-argus-ii-bionic-eye-for-sale-in-the-us-video/
- Seiji Kameda, Yuki Hayashida, Hiroki Tanaka, Dai Akita, Atsushi Iwata, and Tetsuya Yagi. A multi-channel current stimulator chip intended for a visual cortical implant. 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, July 2013. http://embc.embs.org/files/2013/3260_FI.pdf
- Eric E. Thomson, Rafael Carra, and Miguel A.L. Nicolelis. Perceiving invisible light through a somatosensory cortical prosthesis. Nature Communications, 4, February 2013. DOI: 10.1038/ncomms2497.
- M. Sanna, F. Di Lella, M. Guida, and P. Merkus. Auditory brainstem implants in NF2 patients: results and review of the literature. Otology and Neurotology, 33:154-164, 2012.
- P. M. Johnstone, K. R. Yeager, and E. Noss. Spatial hearing in a child with auditory neuropathy spectrum disorder and bilateral cochlear implants. International Journal of Audiology, 52:400-408, 2013.
- Leigh R. Hochberg, Mijail D. Serruya, Gerhard M. Friehs, Jon A. Mukand, Maryam Saleh, Abraham H. Caplan, Almut Branner, David Chen, Richard D. Penn, and John P. Donoghue. Neuronal ensemble control of prosthetic devices by a human with tetraplegia. Nature, 442:164-171, 2006.
- Leigh R. Hochberg, Daniel Bacher, Beata Jarosiewicz, Nicolas Y. Masse, John D. Simeral, Joern Vogel, Sami Haddadin, Jie Liu, Sydney S. Cash, Patrick van der Smagt, and John P. Donoghue. Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. Nature, 485:372-375, 2012.
- Visual phenomena and optical illusions. Internet. http://www.michaelbach.de/ot/http://www.michaelbach.de/ot/
- Best illusion of the year contest. Internet. http://illusionoftheyear.com/http://illusionoftheyear.com
- McGurk effect. Wikipedia. http://en.wikipedia.org/wiki/McGurk_effect
- Tactile illusions. Wikipedia. http://en.wikipedia.org/wiki/Tactile_illusion

15.3.4.18. Modelling user behaviour

- T9 (predictive text). Wikipedia. http://en.wikipedia.org/wiki/T9_(predictive_text)
- Dasher. Wikipedia. http://www.inference.phy.cam.ac.uk/dasher/

- Deep brain stimulation. Wikipedia. http://en.wikipedia.org/wiki/Deep_brain_stimulation
- Brain-computer interface. Wikipedia. https://en.wikipedia.org/wiki/Brain-computer_interface
- Dwight W. Read. How Culture Makes Us Human. Left Coast Press, Walnut Creek, CA, 2012.
- Human communication. Wikipedia. http://en.wikipedia.org/wiki/Human_communication
- Philosophy of mind. Wikipedia. http://en.wikipedia.org/wiki/Philosophy_of_mind
- Nadja Reissland. Neonatal imitation in the first hour of life: Observations in rural nepal. Developmental psychology, 24:464-469, 1988.
- Andrew N. Meltzoff and Moore M. Keith. Imitation in newborn infants: Exploring the range of gestures imitated and the underlying mechanisms. Developmental psychology, 25:954-962, 1989.
- Emese Nagy, Karen Pilling, Hajnalka Orvos, and Peter Molnar. Imitation of tongue protrusion in human neonates: Specificity of the response in a large sample. Developmental psychology, 2012. doi: 10.1037/a0031127.
- Susan S. Jones. Imitation and empathy in infancy. Cognition, Brain, Behavior, 13:391-413, 2009.
- Paralanguage. Wikipedia. https://en.wikipedia.org/wiki/Paralanguage
- Prosody. Wikipedia. https://en.wikipedia.org/wiki/Prosody_(linguistics)
- Paul Ekman and Wallace F. Friesen. Unmasking the Face. Malor Books, Los Altos, CA, 2003.
- Facial action coding system. Wikipedia. http://en.wikipedia.org/wiki/Facial_Action_Coding_System
- A. A. Marsh, H. A. Elfenbein, and N. Ambady. Nonverbal 'accents': cultural differences in facial expressions of emotion. Psychology Science, 14:373-376, 2003.
- A. A. Marsh, H. A. Elfenbein, and N. Ambady. Separated by a common language: Nonverbal accents and cultural stereotypes about americans and australians. J. of Cross Cultural Psychology, 38:284-301, 2007.
- H. Hill and A. Johnston. Categorising sex and identity from the biological motion of faces. Current Biology, 11:880-885, 2001.
- Rachel Adelson. Detecting deception. Monitor on Psychology, 37:70, 2004.
- Paul Ekman. eMETT (Ekman Micro Expression Training Tool) Lite. Internet.https://paulekman.com/learn/content/training-tools-0
- Mihaly Csikszentmihalyi. Finding Flow: The Psychology of Engagement With Everyday Life. Basic Books, New York, NY, 1998.
- Flow (in psychology). Wikipedia. http://en.wikipedia.org/wiki/Flow_(psychology)
- A european network of excellence in social signal processing. Wikipedia. http://sspnet.eu/
- SSPNET dataset. Wikipedia. http://sspnet.eu/category/sspnet_resource_categories/resource_type_classes/dataset/
- SSPNET tools. Wikipedia. http://sspnet.eu/category/sspnet_resource_categories/resource_type_classes/tool/
- Moataz El Ayadi, Mohamed S. Kamel, and Fakhri Karray. Survey on speechemotion recognition: Features, classification schemes, and databases. Pattern Recognition, 44:572-587, 2011.
- Shashidhar G. Koolagudi and K. Sreenivasa Rao. Emotion recognition from speech: a review. International Journal of Speech Technology, 15:99-117, 2012.

- Anton Batliner, Bj¨orn Schuller, Dino Seppi, Stefan Steidl, Laurence Devillers, Laurence Vidrascu, Thurid Vogt, Vered Aharonson, and Noam Amir. The automatic recognition of emotions by speech. Speech Communication, 53:1062-1087, 2011.
- Principal component analysis. Wikipedia. https://en.wikipedia.org/wiki/Principal_component_analysis.
- Jonathon Shlens. A tutorial on principal component analysis. Technical report, New York University Systems Neurobiology Laboratory, 2009. http://www.snl.salk.edu/~ shlens/pca
- J. M. Saragih, S. Lucey, and J. F. Cohn. Deformable model fitting by regularized landmark mean-shift. International Journal of Computer Vision, 91(2):200-215, 2011.
- T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(6):681-685, 2001.
- C. M. Yang, Z. S. Lin, C. L. Hu, Y. S. Chen, L. Y. Ke, and Y. R. Chen. A novel dynamic sensing of wearable digital textile sensor with body motion analysis. In Proc. of the IEEE Conf. on Eng. Med. Biol. Soc., pages 4898-4901, 2010.
- N. N. Y. Chu, Chang-Ming Yang, and Chih-Chung Wu. Game interface using digital textile sensors, accelerometer and gyroscope. IEEE Transactions on Consumer Electronics, 58:184-189, 2012.
- Linear-quadratic regulator. Wikipedia. http://en.wikipedia.org/wiki/Linear-quadratic_regulator
- I. Szita, B. Tak'acs, and A. L"orincz. Epsilon-MDPs: Learning in varying environments. Journal of Machine Learning Research, 3:145-174, 2003.
- A. Lorincz, I. Polik, and I. Szita. Event-learning and robust policy heuristics. Cognitive Systems Research, 4:319-337, 2003.
- I. Szita and A. Lorincz. Optimistic initialization and greediness lead to polynomial time learning in factored MDPs. In Leon Bottou and Michael Littman, editors, International Conference on Machine Learning (ICML 2009), pages 1001-1008, Montreal, June 2009. Omnipress.
- D. J. Ward and D. J. C. MacKay. Fast hands-free writing by gaze direction. Nature, 418:838, 2002.
- Prediction by partial matching. Internet. http://en.wikipedia.org/wiki/Prediction_by_partial_matching
- G. Hevizi, M. Biczo, B. Poczos, Z. Szabo, B. Takacs, and A. Lorincz. Hidden Markov model finds behavioral patterns of users working with a headmouse driven writing tool. In Int. Joint Conf. on Neural Networks 2004, 2004. IEEE Catalog Number: 04CH37541C, Paper No. 1268.
- Viola-Jones object detection framework. Internet. http://en.wikipedia.org/wiki/Viola-Jones_object_detection_framework
- N. L. Roux, Y. Bengio, P. Lamblin, Joliveau M., and B. K'egl. Learning the 2-D topology of images. Advances in Neural Information Processing Systems, pages 841-848, 2007. http://books.nips.cc/papers/files/nips20/NIPS2007_0925.pdf
- A. Lorincz and D. Takacs. Agi architecture measures human parameters and optimizes human performance. In J. Schmidhuber, K.R. Th'orisson, and M. Looks, editors, Conference on Artificial General Intelligence (AGI-2011), volume 6830 of Lecture Notes in in Artificial Intelligence, pages 321-326, Berlin Heidelberg, 2011. Springer-Verlag.
- Hassan K. Khalil. Nonlinear Systems. Prentice Hall, 3rd edition, 2001.
- Richard S. Sutton and Andrew G. Barto. Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA, 1998.
- I. Szita and A. Lorincz. The many faces of optimism: a unifying approach. In Andrew McCallum and Sam Roweis, editors, International Conference on Machine Learning (ICML 2008), pages 1048-1055, Helsinki, Finland, 2008. Omnipress.

- B. Martinez, M. F. Valstar, X. Binefa, and M. Pantic. Local evidence aggregation for regression based facial point detection. IEEE Transactions on Pattern Analysis and Machine Intelligence. 35(5): pp. 1149 - 1163, 2013., 2010. http://sspnet.eu/2011/03/aam-fpt-facial-point-tracker/
- Jason Saragih et .al and Multicomp Lab. Multisense. Internet, 2013.https://vhtoolkit.ict.usc.edu/plugins/viewsource/viewpagesrc.action?pageId=13566443
- B. Jiang, M.F. Valstar, and M. Pantic. Laud: Action unit detection using sparse appearance descriptors in space-time video volumes. in Proc. IEEE Int'l Conf. on Automatic Face and Gesture Recog. (FG'11), 2011.
- http://ibug.doc.ic.ac.uk/resources/laud-programme-20102011/
- B. Jiang, M.F. Valstar, and M. Pantic. Taud: Action unit detection using sparse appearance descriptors in space-time video volumes. in IEEE Int fl. Conf. Face and Gesture Recognition (FG f11), March 2011., 2011.http://ibug.doc.ic.ac.uk/resources/temporal-based-action-unit-detection
- Kinect for windows. Internet, 2013. .http://www.microsoft.com/enus/kinectforwindows/develop/developer-downloads.aspx

15.3.4.19. MAS, Scheduling, Cooperation Basics

- L. Busoniu, R. Babuska, and B. De Schutter, A Comprehensive Survey of Multiagent Reinforcement Learning, IEEE Trans. on Systems, Man, and Cybernetics, Part C: App. and Reviews, vol. 38, no. 2, pp. 156 172, Mar. 2008.
- Y. Labrou, T. Finin, and Y. Peng, Agent communication languages: the current landscape, IEEE Intell. Systems and their App., vol. 14, no. 2, pp. 45 52, Apr. 1999.
- E. Grislin-Le Strugeon, A. Anli, and E. Adam, A Methodology to Bring MAS to Information Systems, in Agent- Oriented Information Systems IV, vol. 4898, M. Kolp, B. Henderson-Sellers, H. Mouratidis, A. Garcia, A. Ghose, and P. Bresciani, Eds. Springer Berlin / Heidelberg, 2008, pp. 90-104.
- C. Georgousopoulos and O. F. Rana, An approach to conforming a MAS into a FIPA-compliant system, in Proc. of the first int. joint conf. on Autonomous agents and multiagent systems: part 2, New York, NY, USA, 2002, pp. 968-975.
- M. Wooldridge, An Introduction to MultiAgent Systems. John Wiley and Sons, 2008.
- W. Ren, R. W. Beard, and E. M. Atkins, A survey of consensus problems in multi-agent coordination, in Proc. of the American Control Conf., 2005, pp. 1859 - 1864 vol. 3.
- R. Olfati-Saber, J. A. Fax, and R. M. Murray, Consensus and Cooperation in Networked Multi-Agent Systems, Proc. of the IEEE, vol. 95, no. 1, pp. 215 -233, Jan. 2007.
- V. R. Lesser, Cooperative multiagent systems: a personal view of the state of the art, IEEE Trans. on Knowledge and Data Eng., vol. 11, no. 1, pp. 133 - 142, Feb. 1999.
- G. Weiss, Ed., Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. The MIT Press, 1999.
- D. Srinivasan, M. C. Choy, and R. L. Cheu, Neural Networks for Real-Time Traffic Signal Control, IEEE Trans. on Intell. Transportation Systems, vol. 7, no. 3, pp. 261 - 272, Sep. 2006.
- R. H. Bordini, J. F. Hübner, and M. Wooldridge, Programming Multi-Agent Systems in AgentSpeak using Jason. John Wiley and Sons, 2007.
- R. Axelrod, The complexity of cooperation: Agent-based models of competition and collaboration. Princeton University Press, 1997.
- D. Greenwood, M. Lyell, A. Mallya, and H. Suguri, The IEEE FIPA approach to integrating software agents and web services, in Proc. of the 6th int. joint conf. on Autonomous agents and multiagent systems, New York, NY, USA, 2007, pp. 276:1-276:7.

- R. A. Flores-Mendez, "Towards a standardization of multi-agent system framework," Crossroads, vol. 5, no. 4, pp. 18-24, Jun. 1999.
- S. D. Ramchurn, D. Huynh, and N. R. Jennings, "Trust in multi-agent systems," Knowl. Eng. Rev., vol. 19, no. 1, pp. 1-25, Mar. 2004.
- E. Semsar-Kazerooni and K. Khorasani, "A game theory approach to multi-agent team cooperation," in American Control Conference, 2009. ACC-09, 2009, pp. 4512 -4518.
- S. C. Hayden, C. Carrick, and Q. Yang, "Architectural Design Patterns for Multiagent Coordination," in Proc. of the 3rd Int. Cnf. on Autonomous Agents, AGENTS-99, 1999.
- G. Lu, J. Lu, S. Yao, and J. Yip, "A Review on Computational Trust Models for Multi-agent Systems," The Open Information Science J., vol. 2, no. 2, pp. 18-25, Mar. 2009.
- G. F. Luger, Artificial Intelligence: Structures and Strategies for Complex Problem Solving, 6th ed. USA: Addison-Wesley Publishing Company, 2008.
- D. Ouelhadj and S. Petrovic, "A survey of dynamic scheduling in manufacturing systems," J Sched, vol. 12, no. 4, pp. 417-431, Aug. 2009.
- A. Rawal, P. Rajagopalan, and R. Miikkulainen, "Constructing competitive and cooperative agent behavior using coevolution," in 2010 IEEE Symp. on Comp. Intelli. and Games (CIG), 2010, pp. 107 -114.
- D. Xiang, Y. Du, L. Yi, and K. Li, "Coordination and Communication among Topic Specific Search Agents," in Third Int. Conf. on Natural Computation, 2007. ICNC 2007, 2007, vol. 4, pp. 703 -707.
- F.-S. Hsieh, "Developing cooperation mechanism for multi-agent systems with Petri nets," Eng. App. of Artificial Intelligence, vol. 22, no. 4 5, pp. 616-627, Jun. 2009.
- K. S. Decker, "Distributed problem-solving techniques: A survey," IEEE Trans. on Systems, Man and Cybernetics, vol. 17, no. 5, pp. 729-740, 1987.
- F. Buccafurri, D. Rosaci, G. M. L. Sarn , and L. Palopoli, "Modeling cooperation in multi-agent communities," Cognitive Systems Research, vol. 5, no. 3, pp. 171-190, Sep. 2004.
- G. Weiss, M. Nickles, M. Rovatsos, and F. Fischer, "Specifying the Intertwining of Cooperation and Autonomy in Agent-based Systems," J. of Networks and Computer App., vol. 29, p. 2007.
- R. G. Smith, "The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver," IEEE Trans. on Comp., vol. C-29, no. 12, pp. 1104-1113, 1980.
- A. Arenas, J. Camacho, J. A. Cuesta, and R. J. Requejo, "The joker effect: Cooperation driven by destructive agents," J. of Theor. Biology, vol. 279, no. 1, pp. 113-119, Jun. 2011.
- E. H. Durfee, V. R. Lesser, and D. D. Corkill, "Trends in cooperative distributed problem solving," IEEE Trans. on Knowledge and Data Eng., vol. 1, no. 1, pp. 63-83, 1989.
- R. G. Reynolds, Z. Kobti, T. A. Kohler, and L. Y. L. Yap, "Unraveling ancient mysteries: reimagining the past using evolutionary computation in a complex gaming environment," IEEE Trans. on Evolutionary Comp., vol. 9, no. 6, pp. 707 - 720, Dec. 2005.
- G. Confessore, S. Giordani, and S. Rismondo, "A market-based multi-agent system model for decentralized multi-project scheduling," Ann Oper Res, vol. 150, no. 1, pp. 115-135, Mar. 2007.
- M. P. Wellman, W. E. Walsh, P. R. Wurman, and J. K. MacKie-Mason, "Auction Protocols for Decentralized Scheduling," Games and Economic Behavior, vol. 35, no. 1-2, pp. 271-303, Apr. 2001.
- D. M. Reeves, M. P. Wellman, J. K. MacKie-Mason, and A. Osepayshvili, "Exploring bidding strategies for market-based scheduling," Decision Support Systems, vol. 39, no. 1, pp. 67-85, Mar. 2005.
- D. E. Simon, An Embedded Software Primer, First Edition. Addison-Wesley Professional, 1999.

- K. Yaghmour, J. Masters, G. Ben-Yossef, and P. Gerum, Building Embedded Linux Systems, 2nd Ed. O'Reilly Media, 2008.
- C. Hamacher, Z. Vranesic, S. Zaky, and N. Manjikian, Computer Organization and Embedded Systems, 6th ed. McGraw-Hill Science/Engineering/Math, 2011.
- H. Kopetz, GENESYS - A Cross-Domain Architecture for Dependable Embedded Systems, in 2011 5th Latin-American Symposium on Dependable Computing (LADC), 2011, pp. 1 -6.
- E. A. Lee and S. A. Seshia, Introduction to Embedded Systems: A Cyber-Physical Systems Approach, 1ST ed. Lulu.com, 2011.
- E. White, Making Embedded Systems: Design Patterns for Great Software. O'Reilly Media, 2011.

15.3.4.20. Sensor networks

- K. R. Fowler, The Future of Sensors and Sensor Networks, Survey Results Projecting the Next 5 Years, Proc. of Sensors Applications Symposium, SAS 2009, New Orleans, LA, USA, 17-19 Feb. 2009, pp. 1 - 6
- P. Gajbhiye and A. Mahajan, A Survey of Architecture and Node Deployment in Wireless Sensor Network, Proc. of Int. Conf. on Applications of Digital Information and Web Technologies, 2008. ICADIWT, Ostrava, Czech Republic, August 4 - 6 2008, pp 426-430.
- P. Györke and B. Pataki, Energy aware measurement scheduling in WSNs used in AAL applications, Accepted to IEEE Trans. on Instrumentation and Measurement, 2013..
- V. Jelicic, T. Razov, D. Oletic, M. Kuri and V. Bilas, MasliNET: A Wireless Sensor Network based Environmental Monitoring System, Proc. on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2011, 23-27 May 2011, Opatija Croatia, pp. 150 - 155.
- R.V. Kulkarni, A. Förster, and G.K. Venayagamoorthy, , Computational Intelligence in Wireless Sensor Networks: A Survey, Surveys and Tutorials, IEEE , vol.13, no.1, pp.68-96, First Quarter 2011
- W. Lin, D. Li, Y. Tan, J. Chen and T. Sun, Architecture of Underwater Acoustic Sensor Networks: A Survey, Proc. of First International Conference on Intelligent Networks and Intelligent Systems, 2008
- Ch. Lin, Y. He, Ch. Peng and L. T. Yang, A Distributed Efficient Architecture for Wireless Sensor Networks, Proc. of 21st International Conference on Advanced Information Networking and Applications Workshops, 2007, AINAW '07, Niagara Falls, Ontario, Canada 21-23 May 2007, pp 429-434
- G.H. Raghunandan and B.N. Lakshmi, A Comparative Analysis of Routing Techniques for Wireless Sensor Networks, Proc. of the National Conference on Innovations in Emerging Technology-201, Perundurai, Erode, Tamilnadu, India.17-18. February, 2011.pp.17-22.
- Fakhreddine Karray, Milad Alemzadeh, Jamil Abou Saleh and Mo Nours Arab, Human-Computer Interaction: Overview on State of the Art, Int. J. on Smart Sensing and Intelligent Systems, Vol. 1, No. 1, March 2008,https://en.wikipedia.org/wiki/Seven_stages_of_action

15.3.5. Appendix B. Control questions

15.3.5.1. Ambient intelligence

What is the essence of the Ambient Intelligence (AmI) paradigm? Define the concept of an Intelligent Space! Define the concept of a Smart Home as an information system! What is covered by AAL and AAC abbreviations? What are the special properties of the AmI applications? Characterize the smart home agent taking into consideration its special new problems! What is context? How can you characterize context-aware computing techniques? Elaborate on an example! How can you characterize privacy-aware computing techniques? Elaborate on an example! What is iHCI (Implicit Human Computer Interaction)? Illustrate the answer with an example! List the components of a typical AmI applications?

15.3.5.2. Context

Characterize context-aware system! What are so called AmI scenarios? What is the purpose of the memory prosthesis? What should we expect regarding 'Ambient interfaces for the elderly'? What is the meaning of W5+ (context-aware systems)? List some essential context information in the AmI environment! What processing capabilities are based on the context information? What is context management? When speaking about context information what is a Pull and Push? Characterize the layers of the context processing architecture! What is the context aggregation? Characterize the structured context processing with transformation modules and control processes! What is a context broker and why we need it? What is the purpose of extendble context ontologies? What problems can we encounter in context-aware applications implemented on computationally resource-poor devices? Why it is important to recognize emotions in AmI applications?

15.3.5.3. Architecture

What is hyper reality? What is a channel, a modality, and a multi-modal communication? Compare from the point of view of AmI applications systems with uni(mono)modal and multi-modal HCI! What are the challenges of the AAL applications? What is the principle of 'augmented reality' interfaces? What are the ambient displays?

15.3.5.4. Sensors

What are so called 'serendipitous' sensors? What are the special requirements for picture processing in AmI (AAL) applications? What is the meaning of Activity of Daily Living (ADL)? Why is sensor fusion needed in AmI applications? What are the technical challenges in sensor networks? What is the energy awareness? What is a Kalman-filter model? What is are Sensor Web Enablement Standards (SWE)? What are their components? What are the special properties of the Sensor Web approach? What is SensorML? Characterize the problem of sensor discovery in sensor networks?

15.3.5.5. Control design in AmI

What is the principal problem of the ambient control? What is the principal problem of the (fuzzy) learning in an ambient environment? What is the principle of the Adaptive Online Fuzzy Inference System (AOFIS)? How to identify membership functions form sensory measurement data? How to identify fuzzy rules from sensory measurement data? What is the principle of the online adaptation and 'life-long' learning? What are type-2 fuzzy sets? What is the Foot Print of Uncertainty? What is the primary and secondary membership function in type-2 fuzzy sets? What improvement do we expect from type-2 fuzzy sets? How do we obtain type-2 fuzzy sets? What is the structure of type-2 FLC - Fuzzy Logic Controller? What is the purpose of Karnik-Mendel algoritm? What is the structure of an hierarchical fuzzy system?

15.3.5.6. Planning

What problems must be solved in planning in an AmI applications? Compare the AI planning problem with the general properties of the AmI systems! Why it is important to know, or recognize the user plans? What is the basic principle of the D-HTN? What are the factors influencing the difficulty of the plan recognition problem? What are the problems with the intention recognition in the AmI applications? What is the basic principle of the Probabilistic Horn Abduction? What is the PHATT Probabilistic Hostile Agent Task Tracker? What are the problems when reasoning about the non observable actions from the observed actions? What are the problems when reasoning about the non observable actions from the observed state transitions?

15.3.5.7. Sensor fusion

Characterize the process of building dynamic model of the environment! What basic principles must be taken into account when building a dynamic model of the environment (the world)? What were the assumptions when deriving within the Bayesian scheme the temporal fusion of the single sensor data? Why we cannot call probability the uncertainty describing quantities used by the Dempster-Shafer fusion? What is the difference between the Bayes and Dempster-Shafer fusion? If we have N events, then how do we represent ignorance in the Dempster-Shafer approach? What problems are caused by in the Dempster-Shafer fusion approach by a conflict? How this problem is solved under Bayesian assumptions? How was Dempster-Shafer approach modified by Yager to handle the conflict problem better? What unified conflict handling approach was proposed by Inagaki? Characterize the statically and dynamically weighted Dempster-Shafer approach! Compare the learned sensor fusion approaches! Show that using Yager's rule the combination of 3 events will not be commutative. Show that the modified MOPs of the weighted fusion method sum up to 1 independently from its w weight, if the sum of the original MOPs of the sensor was 1. Show that using Inakagi's unified combination rule with different k parameters the combination of 3 events will be commutative only in the Dempster-Shafer case.

15.3.5.8. Activity

The action series < a b c d a b c d a b c d a b c d a b c d a b c d .... a b c d > having equidistant timing is analyzed. The series consist of 2013 periods of < a b c d>. Give the compression ratio, if the encoding of the time stamps is twice as long as the encoding of the actions.

The granularity used in a daily activity characterization is minute resolution. Give the compression ratio of a two months period using a 27 minute-long episode candidate, which occur every Tuesday and Wednesday exactly starting at 8:13 a.m.

Give the compression ratio of the exercise above if the start time of the episodes in Tuesday and Wednesday is not so strict, it has uniform distribution between 8:00 and 8:30 a.m.

Give the maximal episodes of the action series < a b c d x x c d a b c x x x x x x b c d a b c d .... a b c d > having equidistant timing, if both window capacity and window time are 5. (x is not a real action, it is not important in itself, but < a x b> or < a a x c> could be important.)

Give the important episode candidates using the maximal episodes found in the Exercise above.

15.3.5.9. HCI

How can one influence the brain directly? How can neural prosthetics influence emotions? Explain the differences and the similarities between retinal implants and an active tactile devices! Consider robotic surgery from the point of view of brain-computer interfaces! What is the origin of illusions?

15.3.5.10. Modelling user behaviour

Characterize user modeling in AmI systems! (from the adaptive and personalized point of view) What are personas? How to compute user profiles from personas? What kind of phenomena can be used for emotion estimation? What are the action units of the face? What kind of algorithms are used facial expression estimation? How is the constrained local model built? What is the key assumption? Sketch the key components of an architecture that can model a user! What is an ARX model? How can it be used for control? What is a decision making heuristics? What is the value of an event?