# A Meta-design approach supporting "unwitting" programmers

**Antonio Piccinno**

Interaction, Visualization and Usability Lab

Dipartimento di Informatica, Università di Bari, Italy

# IVU Group

IVU group (Interaction, Visualization and Usability group), supervised by Prof. Maria Francesca Costabile, has the following research interest:

- End-User Development
- Visual Languages
- Multimedia and Multimodal human-computer interation
- Information Visualization
- Visual Data Mining
- Usability Engineering
- E-learning systems Interfaces

# IVU People

| | |
|---|---|
| | |
| Maria Francesca Costabile<br>(Full Professor) | Paolo Buono<br>(Assistant Professor) |
| | |
| Rosa Lanzilotti, Antonio Piccinno, Carmelo Ardito<br>(Research Collaborators) | Francesca Montinaro, Nicholas Caporusso, Adalberto Simeone<br>(PhD Students) |
| | |
| Thomas Pederson<br>(Visiting Researcher,<br>Umea University, Sweden) | Other collaborators and students |

# Collaborations

On topics in this seminar:

- Prof. Piero Mussio
  - Dip. Informatica e COmunicazione (DICO) - University of Milan - Italy
- Dr. Daniela Fogli
  - Dip. di Elettronica per l'Automazione (DEA) - University of Brescia - Italy
- Dr. Loredana Parasiliti Provenza
  - Dip. Informatica e COmunicazione (DICO) - University of Milan - Italy

# Outline

- Difficulties in HCI process
- User diversity
- Co-evolution of user and system
- ICE (Interaction and Co-Evolution) model
- Unwitting software development
- The Software Shaping Workshop design methodology
- Case studies
- Conclusions

# Communication gap

- Users and software designers adopt different reasoning strategies:
  - ***heuristic*** vs. ***algorithmic***
  - ***examples, analogies*** vs. ***deductive abstract tools***
  - ***concreteness*** vs. ***abstraction***

- Users are forced to express their problems in alien "***computerese***" and play the role designers think users have to play

- … but users are ***domain experts*** and ***owners of the problem***

# Problematic aspects affecting HCI

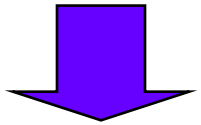
- **User Diversity**
  - users are different in culture, goals, tasks, etc., even in the same community
- **Co-evolution of users and systems**

  "Using the system changes the users, and as they change they will use the system in new ways"

  Nielsen 1993

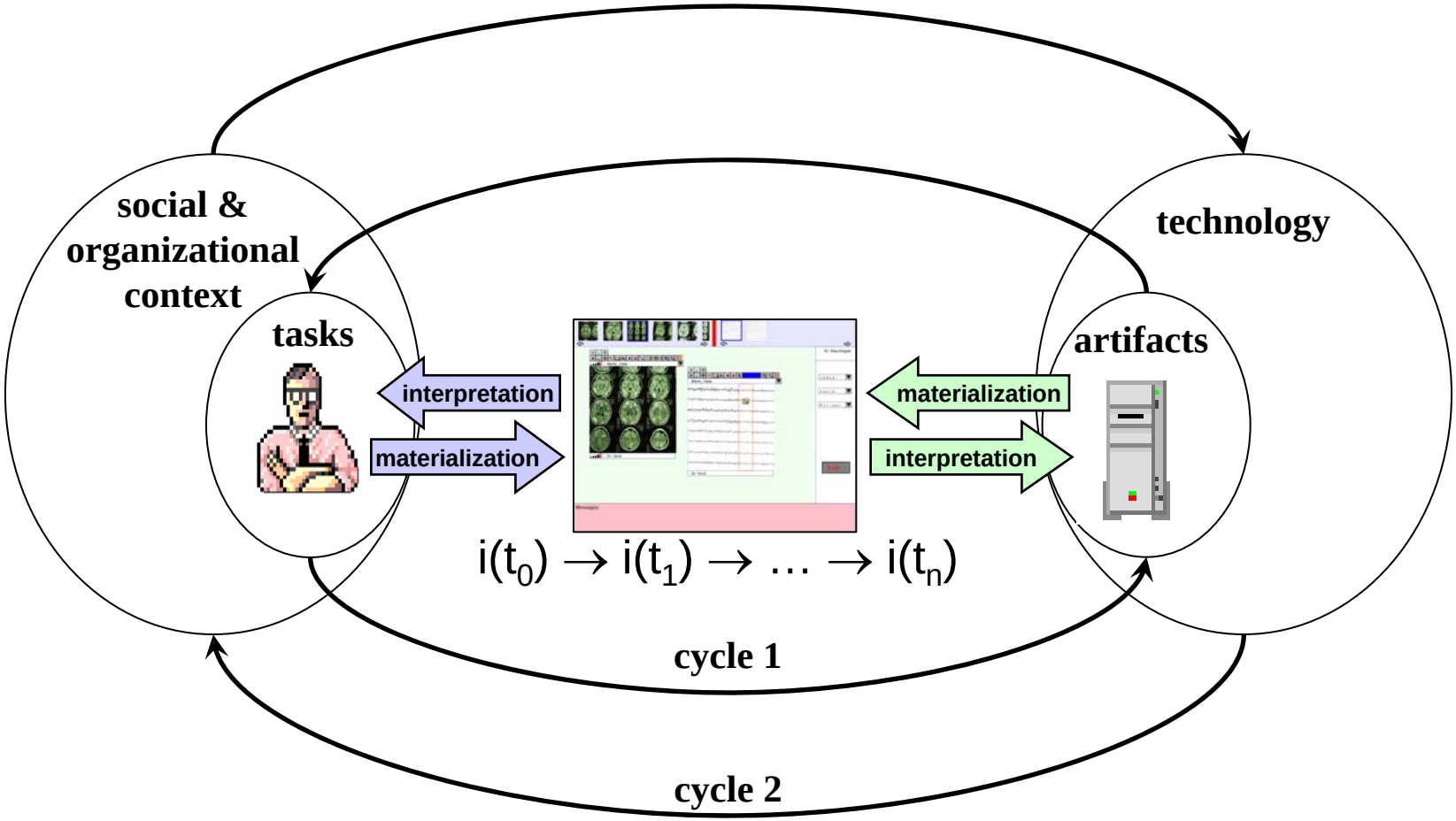
  "The individual is a moving target. Design for the individual of today, and the design will be wrong tomorrow […]. This is because as individuals gain proficiency in usage, they need different interfaces then were required when they were beginners"

  Norman 2005



New model-based approaches to HCI are needed to identify and frame not only the characteristics of the ***interaction process***, but also to consider the ***co-evolution process***

# Interaction and Co-Evolution (ICE) model




Inspired by [Carroll and Rosson 1992][Bourguin et al. 2001]

# Improving HCI

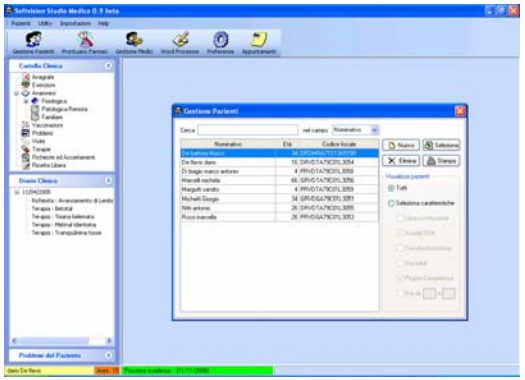
Developing interactive systems:

- that are inspired by the “world” in which end users work (**metaphor**)
- whose **interaction style** (messages + actions) is based on **lexicon and signs** and on **working strategies** that are **familiar** to end users
- Interaction languages exploit end user traditional languages

# “one size fits all” Vs “one-task”

## general purpose tools

- Too difficult to use, to learn and too complex
- General purpose tools are not suitable for end users
- Personalization feature



## one-task tools

- Very few functionalities: only the needed ones
- Easy to use

Tools supporting a limited set of tasks

few functionalities aimed at accomplishing the tasks for a specific purpose or for a specific user community

# Turing Tar Pit

- **Turing Tar Pit:** “Beware of the Turing Tar Pit, in which everything is possible, but nothing of interest is easy.”

- **Inverse of Turing Tar Pit:** “Beware of the over-specialized systems, where operations are easy, but little of interest is possible.”

[G. Fischer 2006]

# Two classes of end-user activities

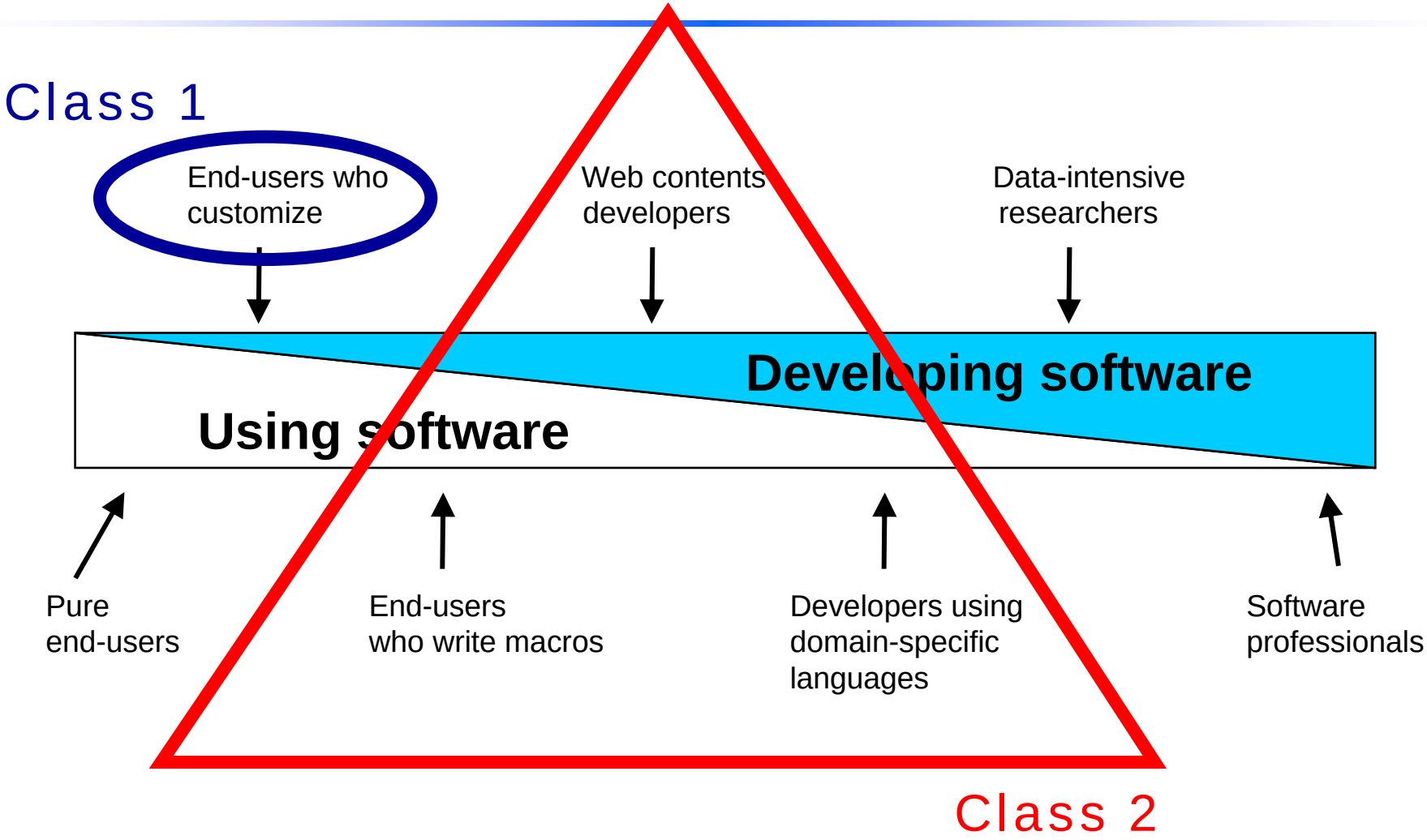
- *Parametrization or customization (by users)*: Class 1
  - Activities allowing users to choose among presentations, behaviours, interaction modalities (foreseen by designers)
- *Tailoring (by users)*: Class 2
  - Activities that require the creation or the modification of a software artifact
    - direct manipulation
    - visual programming
    - programming by example or by demonstration
    - macro
    - Scripting languages

[Costabile, M. F., et al. HCC 2003]

**End User Development** (EUD):

"*End User Development is a set of activities or techniques that allow users of software systems, who are acting as non-professional software developers, at some point to create or modify a software artefact* " [EUD-Net 2003]

# Software-related activities




adapted by

Ye, Y. and Fischer, G. 2007. Designing for Participation in Socio-Technical Software Systems. Proc. HCII 2007 (Beijing, China, Jul. 22-27, 2007). LNCS,Springer, 312-321.

# Unwitting end user programmers

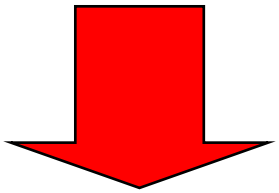
- **Children** playing with a computer game are required sophisticated programming embedded in a motivated activity
- It is perceived as something easy and fun to perform
- It cannot be programming!
- Computer-based authoring tools allow them to construct interactive simulations, animations, and games
  - a lot of emphasis is made on construction
  - they program by construction not by algorithm development

- Programming is not the children goal; playing, constructing and deconstructing are their goal
- *unwitting end user programmers*

[Petre & Blackwell, VL/HCC 2007]

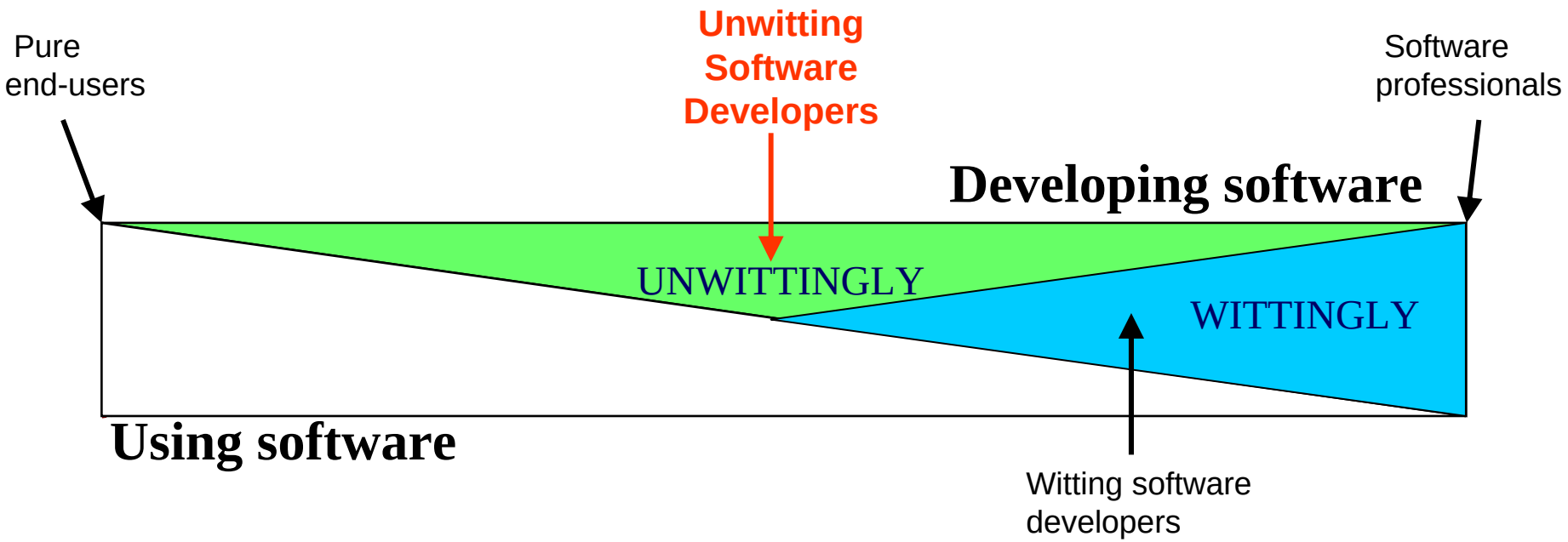
# Domain expert as unwitting programmers

- End users not expert at all in computer science, nor are willing to be, and use computer systems for their daily work activities
- They want software environments easily accessible, that they can “tailor” to their needs, task and habits without being aware of programming



- They are **Unwitting Software Developers**

# From end-users to software professionals



# Designing usable interactive systems

- A design methodology, based on the ICE model, that creates different software environments supporting specific activities
- Each environment provides all and only the tools to perform the desired activities
  - Few functionalities aimed at accomplishing the tasks for a specific purpose or for a specific user community
- When users need more functionalities, the system should evolve

# The Software Shaping Workshop (SSW) methodology

- A useful metaphor for conceptual design: ***artisan workshop***
- In several application domains, different communities of users cooperate to reach a common goal
- Software environments, each devoted to a specific community of users, are organized as virtual workshops, called ***Software Shaping Workshops***

# Two kinds of Software Shaping Workshops

- ***Application workshop*** is a SSW used by a community of end users to perform their daily tasks in a certain domain, it is properly designed for the specific needs of that community of end users

***System workshop*** is a SSW used by a community of experts in the design team to generate and update other workshops, it is properly designed for the specific needs of that community of experts, e.g. software engineers, or HCI experts, or domain experts

# Meta-design

- “Meta-design characterizes objectives, techniques, and processes for creating new media and environments allowing ‘owners of problems’ (that is, end users) to act as designers”

  [Fischer et al. 2004, CACM, Special Issue on End User Development]

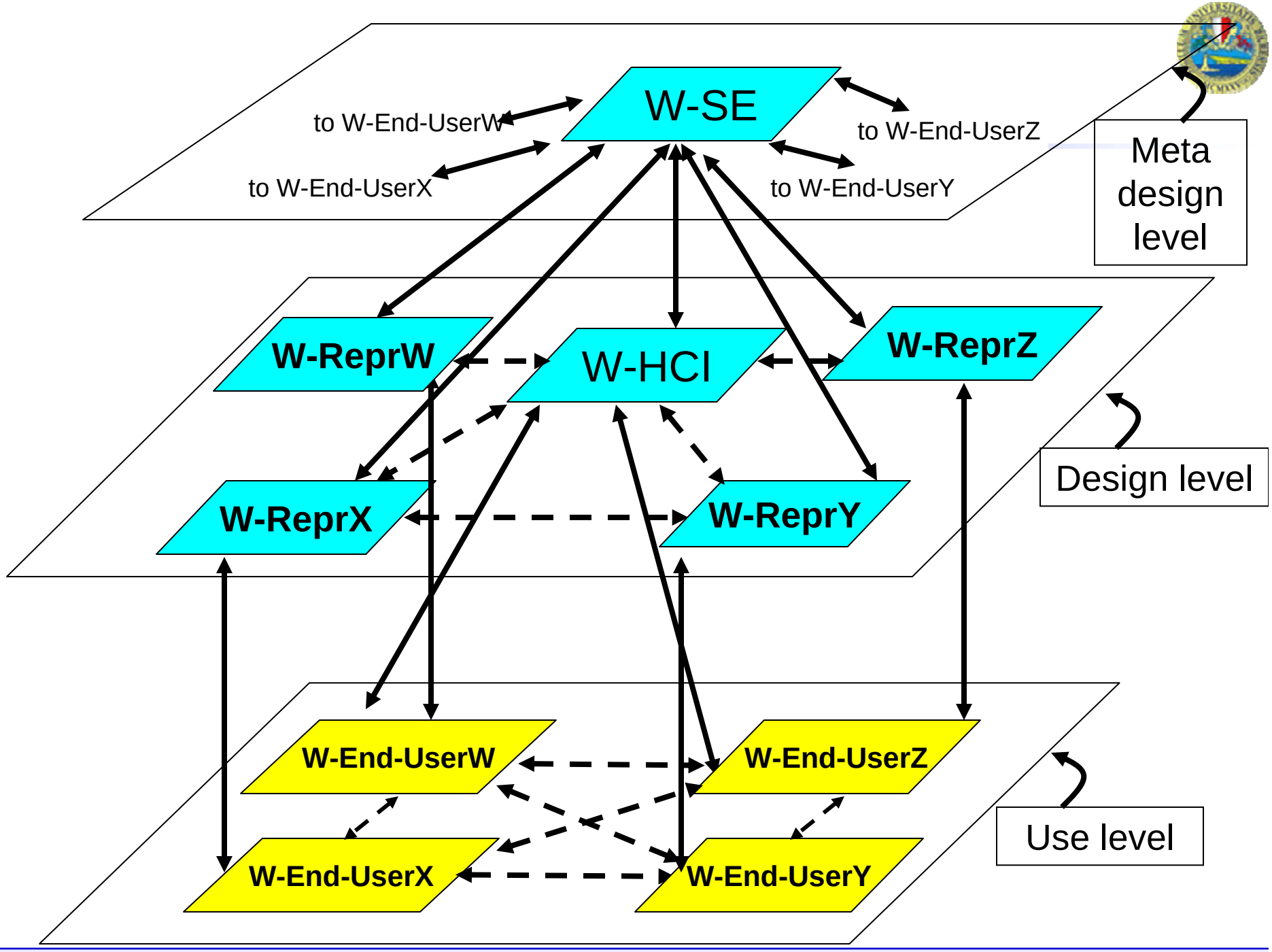
**Our view**:

- A design paradigm that includes end users as active members of the design team and provides all stakeholders in the team with suitable languages and tools to foster their personal and common reasoning about the development of interactive software systems that support end users’ work

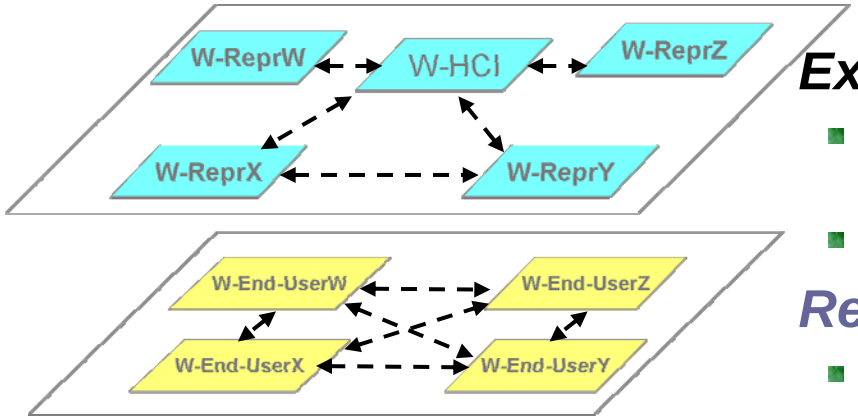
# Workshop network organization

- An interactive system to support the work practice in a given application domain is
  - not a monolithic piece of software…
  - but a ***network of system and application workshops***

- The network levels:
  - The ***meta-design level***: where software engineers use a system workshop to prepare the tools to be used at the successive level
  - The ***design level***: where HCI experts and user representatives cooperate to the design, implementation and validation of application workshops, using system workshops customized to their needs, culture and skills
  - The ***use level***: where end users cooperate to achieve a task using application workshops


W-SE
to W-End-UserW
to W-End-UserZ
to W-End-UserX
to W-End-UserY
Meta design level
W-ReprW
W-HCI
W-ReprZ
W-ReprX
W-ReprY
Design level
W-End-UserW
W-End-UserZ
W-End-UserX
W-End-UserY
Use level

# Supporting co-evolution: the role of communication paths




***Exchange paths:***

- the paths along which the exchanges of data and programs occur
- among the workshops at the same level

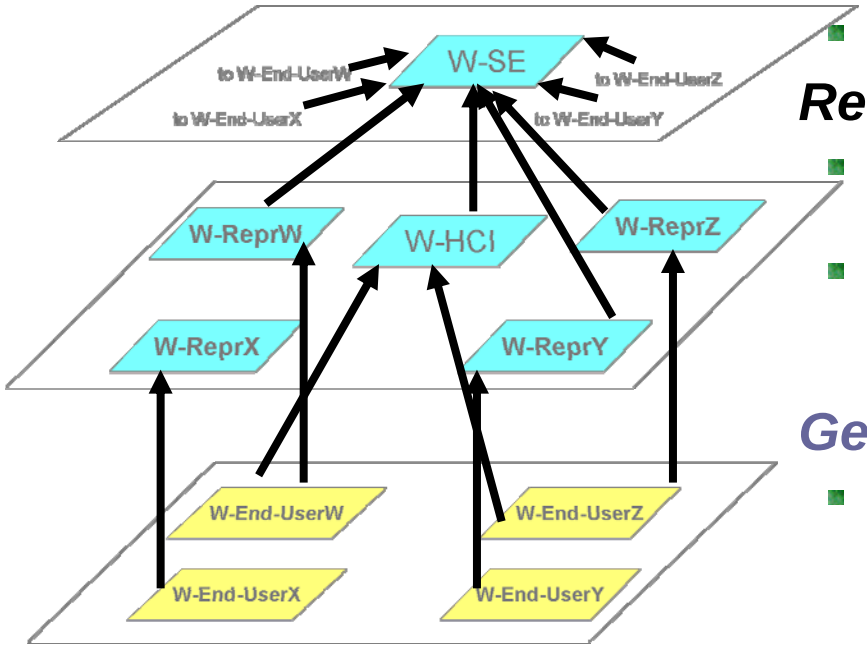
***Request paths:***

- concerned with the communications going from low levels to higher levels
- trigger the co-evolution process, carrying on the feedback from end users (requests for workshop modification or extension)

***Generation paths:***

- represent the activity of using system workshops at a high level to generate, modify or extend workshops to be used at the lower level
- new or evolved workshops are made available to lower levels along such generation paths

# Supporting co-evolution: the role of communication paths




***Exchange paths:***

- the paths along which the exchanges of data and programs occur
- among the workshops at the same level

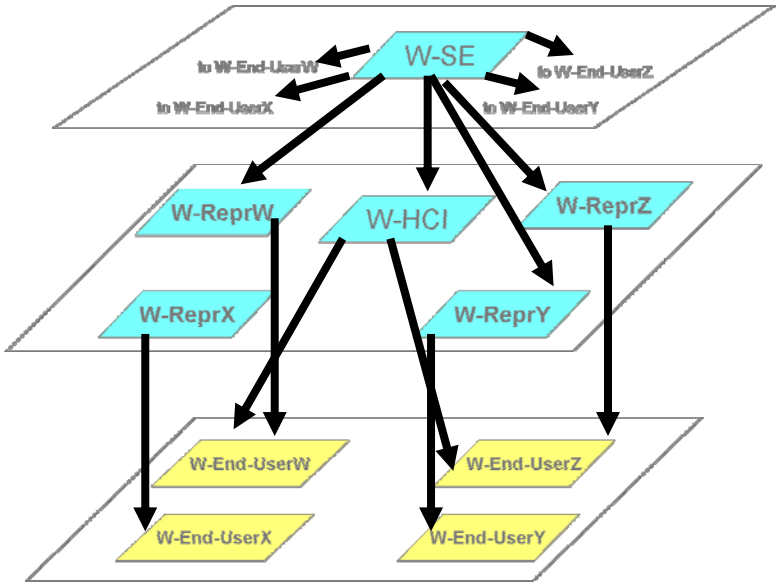
***Request paths:***

- concerned with the communications going from low levels to higher levels
- trigger the co-evolution process, carrying on the feedback from end users (requests for workshop modification or extension)

***Generation paths:***

- represent the activity of using system workshops at a high level to generate, modify or extend workshops to be used at the lower level
- new or evolved workshops are made available to lower levels along such generation paths

# Supporting co-evolution: the role of communication paths




***Exchange paths:***

- the paths along which the exchanges of data and programs occur
- among the workshops at the same level

***Request paths:***

- concerned with the communications going from low levels to higher levels
- trigger the co-evolution process, carrying on the feedback from end users (requests for workshop modification or extension)

***Generation paths:***

- represent the activity of using system workshops at a high level to generate, modify or extend workshops to be used at the lower level
- new or evolved workshops are made available to lower levels along such generation paths

# Achieving co-evolution

- The experts reachable in the network analyze annotations, communicate among them using exchange paths (or request paths, if they in turn refer to the higher level), and agree on a possible solution to the notified problems, thus updating the corresponding workshop (along the generation path)

- Co-evolution: the result of a combination of ***generation, request and exchange activities*** carried out throughout the lifecycle of the SSW network

- ***All workshops*** and ***all stakeholders*** in the network may be potentially involved in the co-evolution process

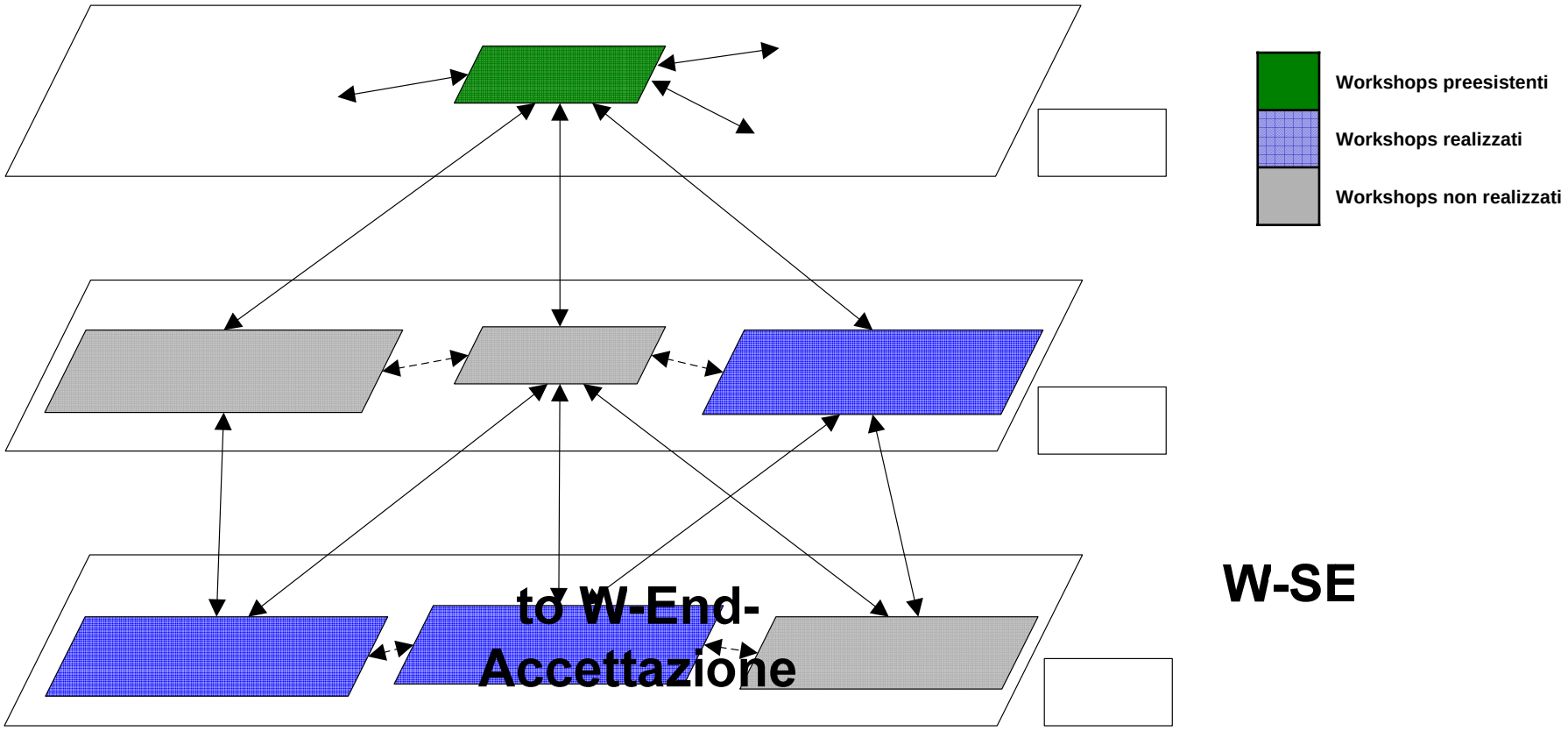
# Case study: patient record

- Paper-based patient records
- Specific patient records for each ward even in the same hospital
- One patient record each patient
- Patient records are composed by modules
- Each module contains specific field for collecting patient data
- Nurse records the patient measurements

# Users and “unwitting” programmers

- *Hospital Management*
  - it defines the needed modules for acceptance of a new patient
- *Reception*
  - it is called to record new patients data
- *Head physician* of the specific ward
  - s/he defines the patient record
- *Nurse*
  - s/he fill in the patient record with data from patient daily measurements

# The SSW network in the case study

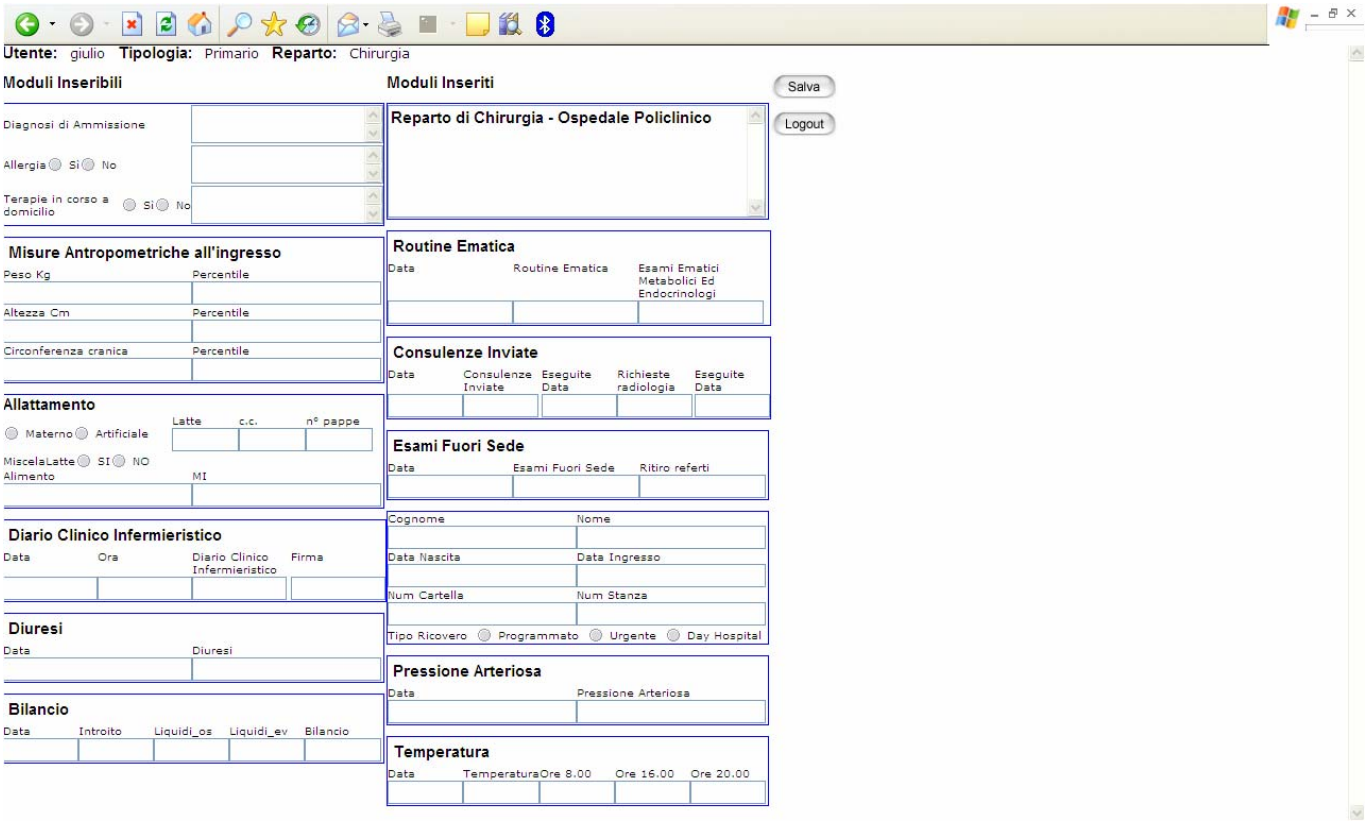


# The SSW network in the prototype

- Meta-Design Level
  - framework Echo2
  - J-Creator, Eclipse
- Design Level
  - “W-ReprDirezioneSanitaria” is used by hospital management to define the reception needed modules
  - “W-ReprPrimario” is used by the various wards head physicians
  - “W-ReprHCI” is used by Human Computer Interaction experts to evaluate the SSW at “use level” usability
- Use Level
  - “W-End-Accettazione” is used by receptionists
  - “W-End-Infermiere” is used by nurses
  - “W-End-Primario” is used by head physicians to examine the patient record

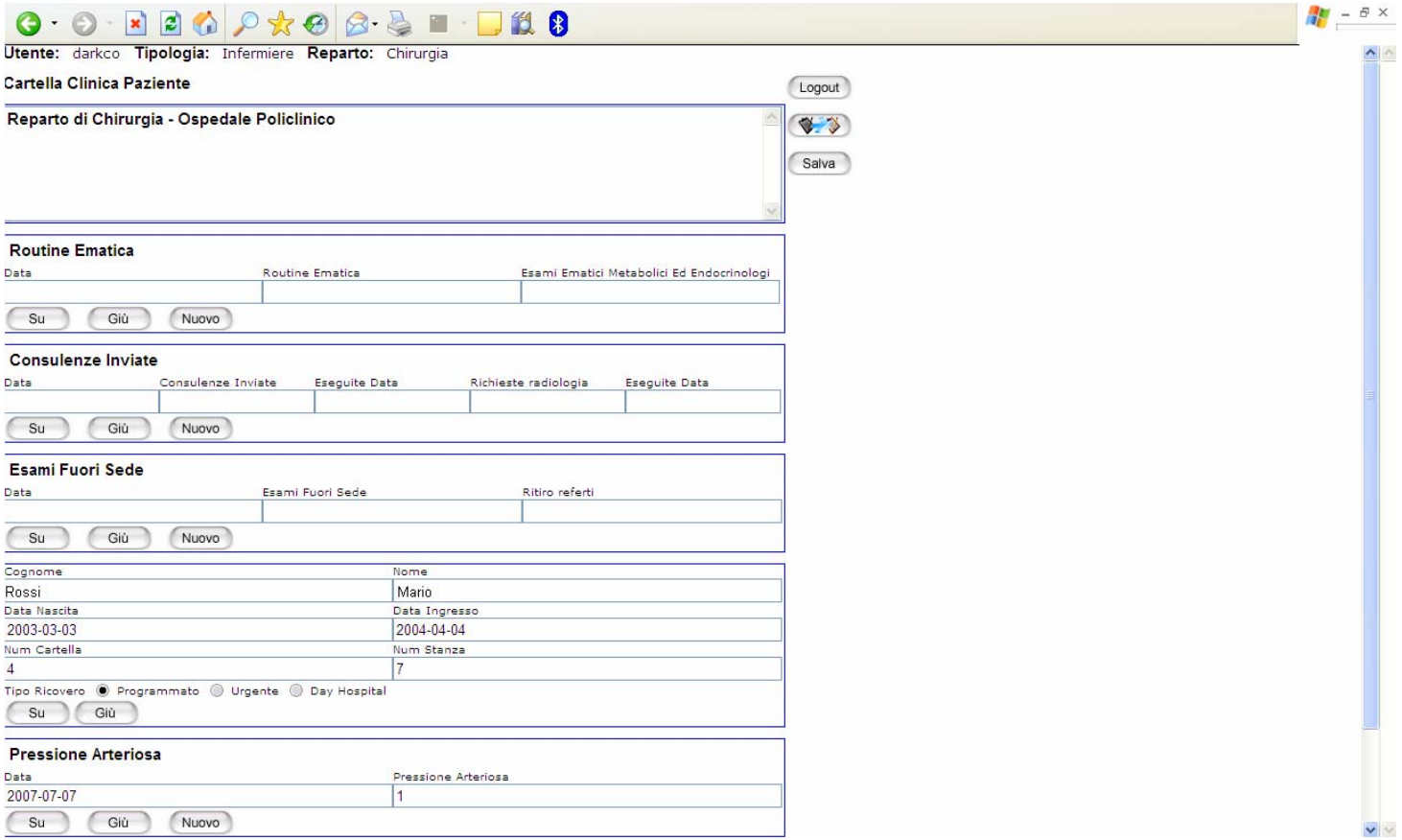
# System Workshop

**W-RapprPrimario** is used by each head physician to create and update the specific application workshops “W-End-Infermiere” and “W-End-Primario” for their own ward



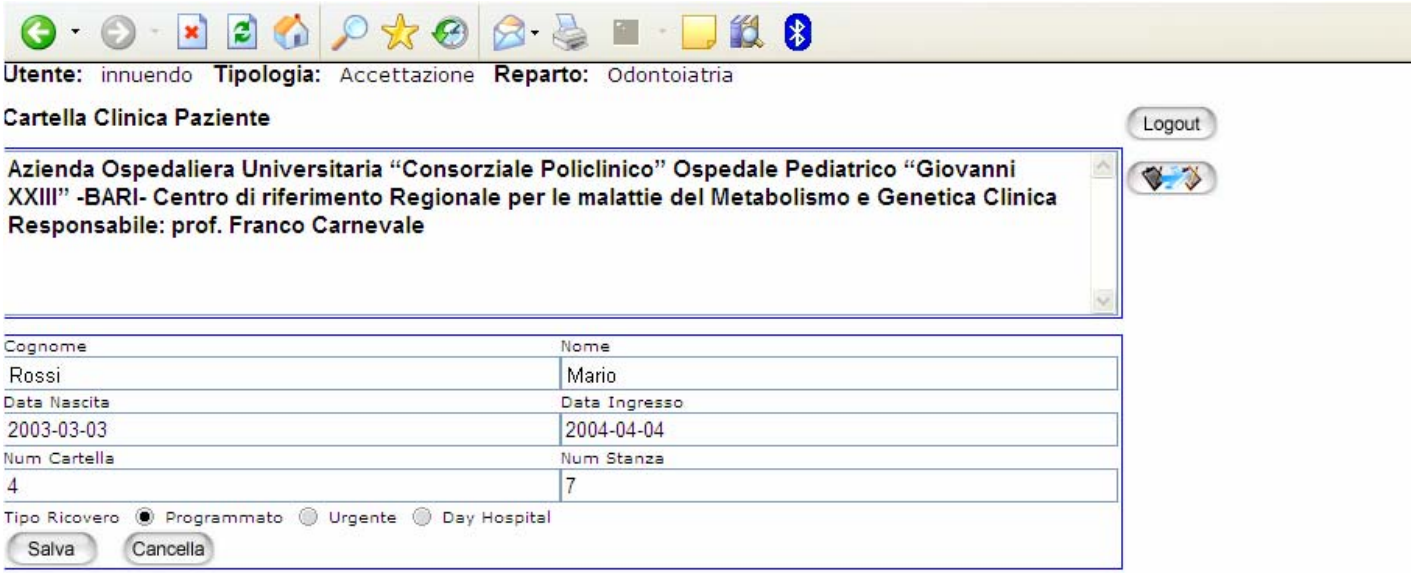
# Application workshop for Nurses

- **W-End-Infermiere** is used by nurses to daily fill in the measurements on the patient (and required in the patient record)

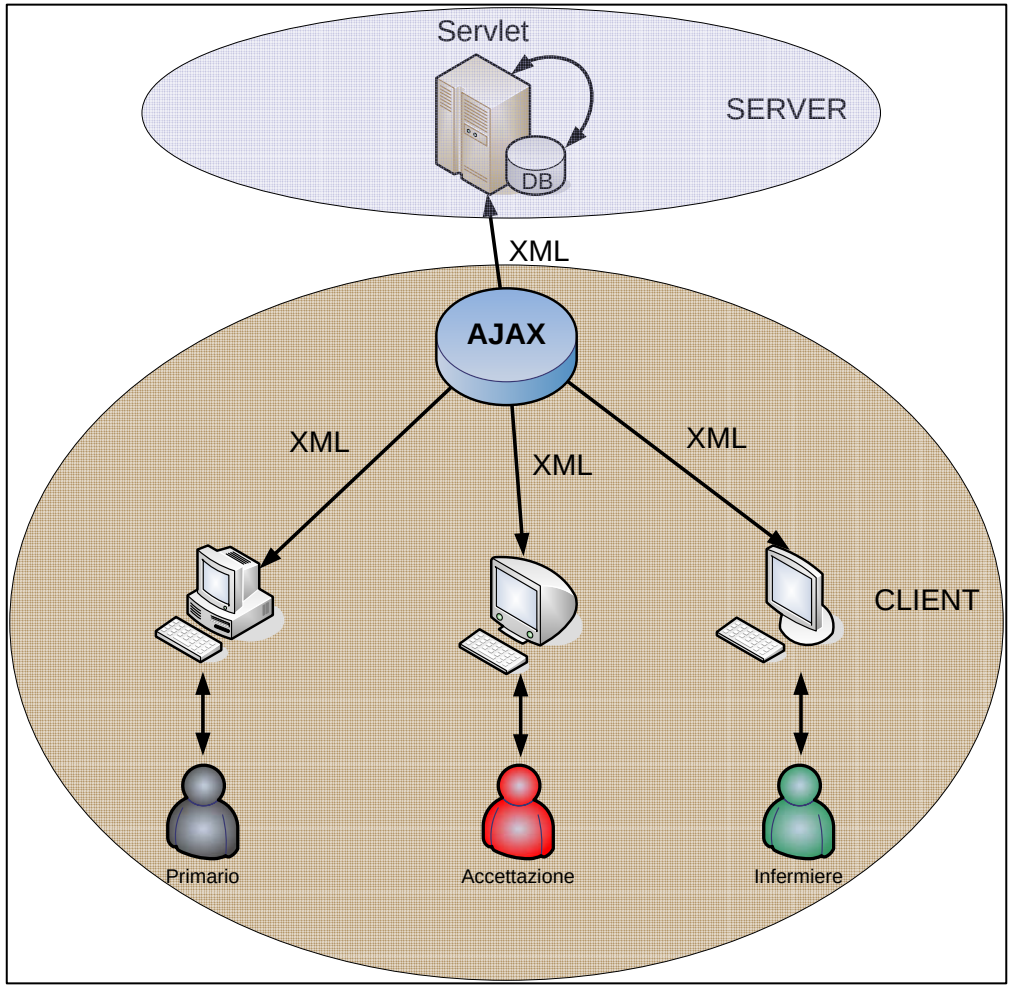


# Application workshop for Reception

- **W-End-Accettazione** is used by hospital receptionist to record new patient data



# Prototype architecture



# Others case studies

- Mechanical engineering field
- Medical domain
- Tourism, Cultural and Geological fields
- ….

# Conclusion

- A model-based methodology aimed at designing interactive systems that address the needs of different communities of users, in which operations are easy to perform
- The SSW methodology bridges the communication gaps among the members of the design team that includes different experts: software engineers, HCI experts, end users as domain experts
- Each expert is a stakeholder that proposes design solutions from his/her perspective
- Each domain expert is an unwitting software developers
- By relying on a novel model of Interaction and Co-Evolution processes, co-evolution of users and systems is supported

dib
Antonio Piccinno
piccinno@di.uniba.it
Thanks!