Why did my laundry turn pink?

Usability and UX in home appliances and consumer electronics

Tiziana Catarci

catarci@diag.uniroma1.it

with F.Leotta, A.Marrella, M.Mecella

SAPIENZA
UNIVERSITÀ DI ROMA

DIPARTIMENTO DI INGEGNERIA INFORMATICA, AUTOMATICA E GESTIONALE "ANTONIO RUBERTI"

CIS SAPIENZA

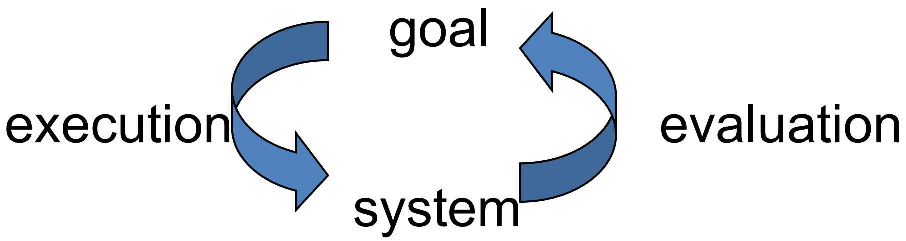
Outline of the talk

- What do we mean by «interaction»
- Passing responsability for task execution from the user to the system: are smart objects good enough?
- How to effectively measure the usability of consumer products

The Interaction

- Communication between user and system (whatever the system is)
  – translations between user and system
- Ergonomics
  – physical characteristics of interaction
- Interaction styles
  – the nature of user/system dialog

Norman's Model



- user establishes the goal
- formulates intention
- specifies actions at interface
- executes action
- perceives system state
- interprets system state
- evaluates system state with respect to goal

goal – intention
what the user would like to become true

task – action
how to achieve it

Some systems are harder to use than others

Gulf of Execution

user's formulation of actions

≠ actions allowed by the system

Gulf of Evaluation

user's expectation of changed system state

≠ actual presentation of this state

Ergonomics

- Study of the physical characteristics of interaction
- Ergonomics good at defining standards and guidelines for constraining the way we design certain aspects of systems
- Examples
  - arrangement of controls and displays

  e.g. controls grouped according to function or frequency of use, or sequentially
  - surrounding environment

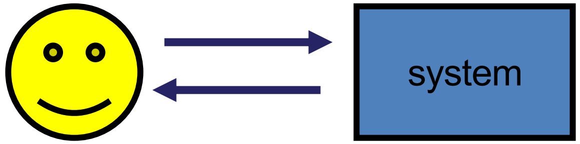
  e.g. seating arrangements adaptable to cope with all sizes of user
  - health issues

  e.g. physical position, environmental conditions (temperature, humidity), lighting, noise,
  - use of colour

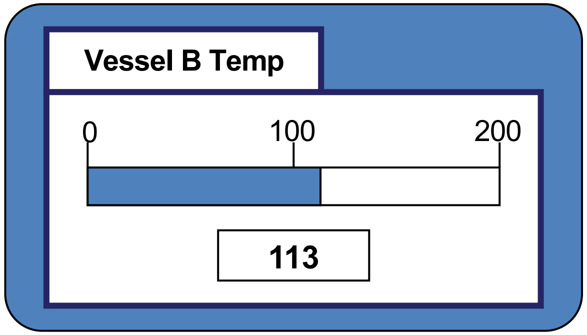
  e.g. use of red for warning, green for okay, awareness of colour-blindness etc.

Industrial interfaces

- Office – direct manipulation
  - user interacts
    with artificial world

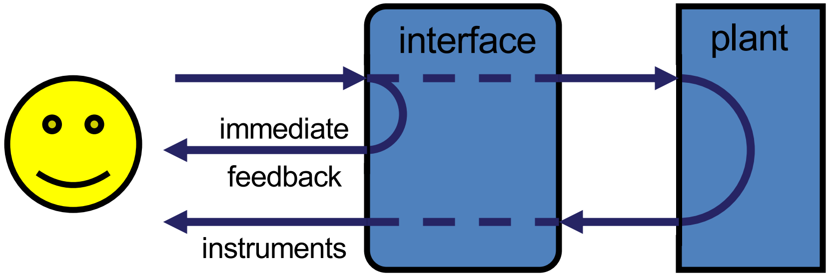


- Industrial – indirect manipulation
  - user interacts
    with real world
    through interface



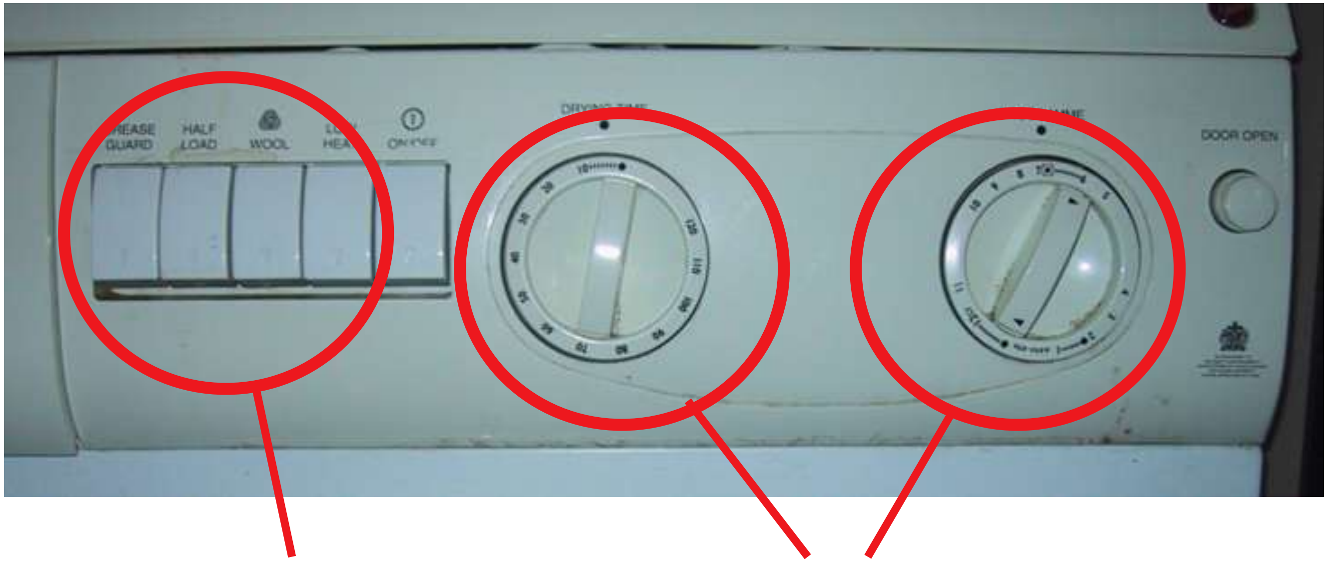
multiple representations of same information

- issues ..
  - feedback
  - delays



Compliant interaction

The importance of the physical interface…



Configuration evident in mechanical buttons

rotary knobs reveal internal state and can be controlled by both user and machine

Usability

- Designing for maximum **usability** (we will not deal with aestethics and enjoyment)
- Usability refers to "*the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.*"
- Principles to support usability
  - Learnability
    - the ease with which new users can begin effective interaction and achieve maximal performance
  - Flexibility
    - the multiplicity of ways the user and system exchange information
  - Robustness
    - the level of support provided to the user in determining successful achievement and assessment of goal-directed behaviour

Relevant sub-characteristics

Task conformance

- degree to which system services support all of the user's tasks
- task completeness; task adequacy

Task migratability

- passing responsibility for task execution between user and system

Observability

- ability of user to evaluate the internal state of the system from its perceivable representation

Familiarity

- how prior knowledge applies to new system
- guessability; affordance

Consistency

- likeness in input/output behaviour arising from similar situations or task objectives

User eXperience (UX)

- Simplicity is key
- Time saving interaction
- Smart systems carrying on the (right) task
  - CE tasks usually very focused and precisely definable
- Cooperative human-system interaction

Internet-of-Things (IoT)

A *system vision* of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with:

- **unique identifiers** (UIDs)
- the **ability to transfer data over a network** *without necessarily* requiring human-to-human or human-to-computer interaction

Smart object

- An object that enhances the interaction with not only people but also with other smart objects
  - smart connected product
  - smart connected thing
  - smart device
- product, asset, other things **embedded with processors, sensors, software and connectivity that allow data to be exchanged between the product and its environment, manufacturer, operator/user, and other products and systems**
  - Connectivity enables some capabilities of the product to exist outside the physical device, in what is known as the product cloud
  - The data collected from this product can be then analyzed to inform decision-making, enable operational efficiencies and continuously improve the performance of the product

A bit of history (1)

- 1982 : a modified Coke vending machine at Carnegie Mellon University was the first Internet-connected appliance, able to report its inventory and whether newly loaded drinks were cold or not
  - https://www.cs.cmu.edu/~coke/history_long.txt
- 1991 : seminal paper *Weiser, Mark: «The Computer for the Twenty-First Century». Scientific American. 265 (3): 94–104*
  - doi:10.1038/scientificamerican0991-94
- 1999 : at the World Economic Forum in Davos, Bill Joy (BSD Unix, vi, Sun Microsystems) envisioned *device-to-device communication* as a part of his "Six Webs" framework
- 1999 : Kevin Ashton of Procter & Gamble, later MIT's Auto-ID Center, conied the term «Internet of things», though he preferred the phrase «Internet for things»
  - https://www.rfidjournal.com/articles/view?4986

A bit of history (2)

- @ 2005 : Arduino (a single-board microcontroller to be used in interactive projects) is invented at the Interaction Design Institute Ivrea (IDII), Italy
- Pervasive / Ubiquitous computing conferences provided scientific/technical advancements to the field
  - Cf. Ubicomp series (since 1999), merged since 2012 with Pervasive Computing, see http://www.ubicomp.org/sc/
    - https://dblp1.uni-trier.de/db/conf/huc/
  - Cf. PerCom series (since 2003)
    - https://dblp1.uni-trier.de/db/conf/percom/

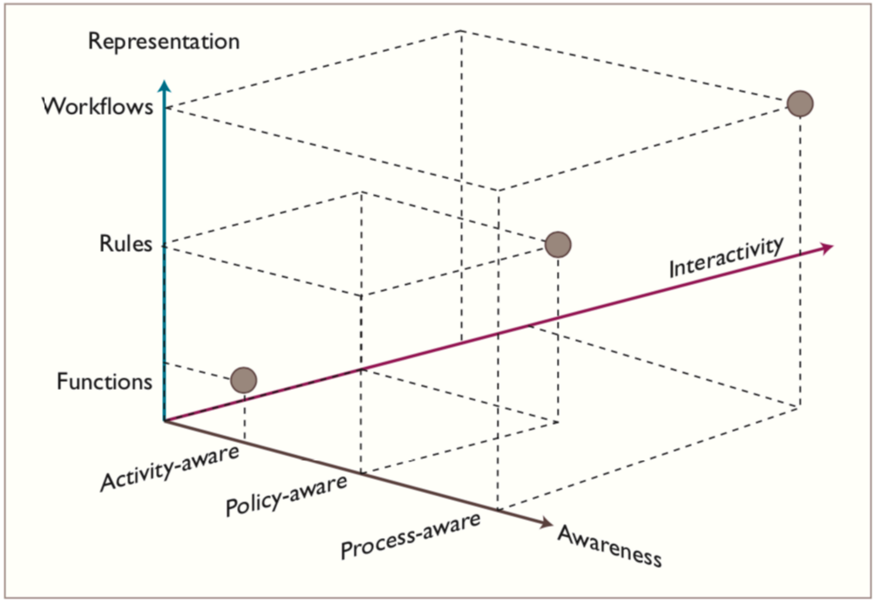
Smart objects as building blocks (1)

G. Kortuem, F. Kawsar, V. Sundramoorthy, D. Fitton. *Smart Objects as Building Blocks for the Internet of Things*. IEEE Internet Computing 14, 1 (January 2010), 44-51

- IoT built from smart objects raises several important research questions in terms of:
  - system architecture, design and development
  - human involvement
- E.g.,
  - What is the right balance for the distribution of functionality between smart objects and the supporting infrastructure?
  - How do we model and represent smart objects' intelligence?
  - What are appropriate programming models?
  - How can people make sense of and ***interact*** with smart physical objects?

Smart objects as building blocks (2)

- Activity-aware objects
- Policy-aware objects
- Process-aware objects



Awareness - the smart object’s ability to understand (that is, sense, interpret, and react to) events and human activities occurring in the physical world
Representation – the smart object’s application and programming model — in particular, programming abstractions
Interaction – the object’s ability to converse with the user in terms of input, output, control, and feedback (including the user awareness of what the smart device is doing)

Smart objects as building blocks (3)

Table 1. Summary of smart-object types.					
	Awareness	**Representation**	**Interaction**	**Augmentation**	**Example application**
Activity-aware object	Activities and usage	Aggregation function	None	Time, state (on/off), vibration	Pay-per-use
Policy-aware object	Domain-specific policies	Rules	Accumulated historical data, threshold warnings	Time, vibration, state, proximity	Health and safety
Process-aware object	Work processes (that is, sequence and timing of activities and events)	Context-driven workflow model	Context-aware task guidance and alerts	Time, location, proximity, vibration, state	Active work guidance

- An activity-aware object understands the world in terms of events and activities directly related with its use
- A policy-aware object can interpret the events and activities wrt predefined organization policies
- A process-aware object understands the organizational processes in which it is involved and can relate real-world events and activities to such processes. Provides users with context-aware guidance about tasks and decisions.

Challenges

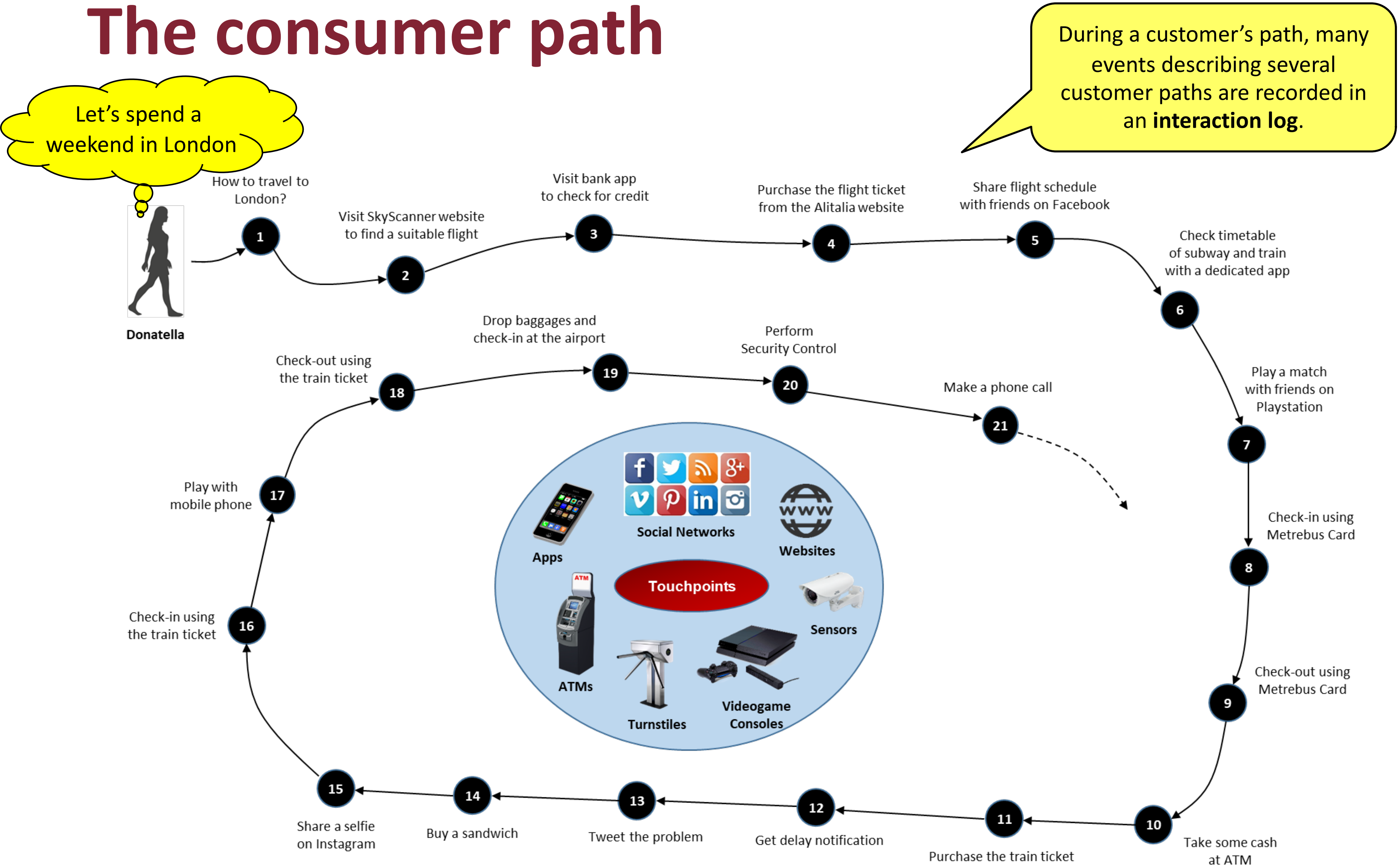
- IoT/smart objects are nowadays the «core» of consumer electronics
  - Their level is going from activity-aware up to policy-aware, and beyond
    - Process-awareness is still to come, subject to research in many fields
  - User and system still need to cooperate to achieve the goal
  - But the interaction is not well thought, therefore the user acceptance and satisfaction is not high ☹
  - There is still the need of improving the consumer product usability

Measuring usability of consumer products

Usability aims at measuring the quality of an interaction

- **Measurable parameters**
  - How long a user takes to perform a task
  - The amount of errors during an interaction
- **Limitations**
  - User evaluations mainly performed in an artificial setting (lab and field studies)
  - *Expensive* and *time-consuming* techniques for observing users over an extended period of time.
  - Limited amount of user tests
- **Tackling these issues**
  - Capture interactions on a **daily basis**
  - Save interactions in dedicated **interaction log files**
  - **Detect** and **understand usability issues** by analysing interaction logs

The consumer path



Interaction Logs

- Include the user actions (from low-level keystrokes to content shared via social media) recorded “in situ” as people interact with UIs of software applications and consumer products, uninfluenced by external observers.
- Easy to capture at scale.
- Technically are multi-set of *execution traces*. Each trace consists of a sequence of *user actions* related to the *single execution of a specific relevant task*.
- Observe even small differences that exist between populations (e.g. demographic, behavior, etc.).

- There is a need to find a way that exploits interaction logs for **identifying exactly what has happened** and **what has gone wrong** during a user’s interaction.

Example of an Interaction Log

case id	event id	properties				
		timestamp	activity	resource		
	35654423	30-12-2010:11.02	register request	Pete		
1	35654424	31-12-2010:10.06	examine thoroughly	Sue		
	35654425	05-01-2011:15.12	check ticket	Mike		
	35654426	06-01-2011:11.18	decide	Sara		
	35654427	07-01-2011:14.24	reject request	Pete		
	35654483	30-12-2010:11.32	register request	Mike		
2	35654485	30-12-2010:12.12	check ticket	Mike		
	35654487	30-12-2010:14.16	examine casually	Pete		
	35654488	05-01-2011:11.22	decide	Sara		
	35654489	08-01-2011:12.05	pay compensation	Ellen		
	35654521	30-12-2010:14.32	register request	Pete		
3	35654522	30-12-2010:15.06	examine casually	Mike		
	35654524	30-12-2010:16.34	check ticket	Ellen		
	35654525	06-01-2011:09.18	decide	Sara		
	35654526	06-01-2011:12.18	reinitiate request	Sara		
	35654527	06-01-2011:13.06	examine thoroughly	Sean		
	35654530	08-01-2011:11.43	check ticket	Pete		
	35654531	09-01-2011:09.55	decide	Sara		
	35654533	15-01-2011:10.45	pay compensation	Ellen		
	35654641	06-01-2011:15.02	register request	Pete	50	...
4	35654643	07-01-2011:12.06	check ticket	Mike	100	...
	35654644	08-01-2011:14.43	examine thoroughly	Sean	400	...
	35654645	09-01-2011:12.02	decide	Sara	200	...
	35654647	12-01-2011:15.44	reject request	Ellen	200	...
	35654711	06-01-2011:09.02	register request	Ellen	50	...
5	35654712	07-01-2011:10.16	examine casually	Mike	400	...
	35654714	08-01-2011:11.22	check ticket	Pete	100	...
	35654715	10-01-2011:13.28	decide	Sara	200	...
	35654716	11-01-2011:16.18	reinitiate request	Sara	200	...
	35654718	14-01-2011:14.33	check ticket	Ellen	100	...
	35654719	16-01-2011:15.50	examine casually	Mike	400	...
	35654720	19-01-2011:11.18	decide	Sara	200	...
	35654721	20-01-2011:12.48	reinitiate request	Sara	200	...
	35654722	21-01-2011:09.06	examine casually	Sue	400	...
	35654724	21-01-2011:11.34	check ticket	Pete	100	...
	35654725	23-01-2011:13.12	decide	Sara	200	...
	35654726	24-01-2011:14.56	reject request	Mike	200	...
	35654871	06-01-2011:15.02	register request	Mike	50	...
6	35654873	06-01-2011:16.06	examine casually	Ellen	400	...
	35654874	07-01-2011:16.22	check ticket	Mike	100	...
	35654875	07-01-2011:16.52	decide	Sara	200	...
	35654877	16-01-2011:11.47	pay compensation	Mike	200	...

case id	trace
1	$\langle a,b,d,e,h \rangle$
2	$\langle a,d,c,e,g \rangle$
3	$\langle a,c,d,e,f,b,d,e,g \rangle$
4	$\langle a,d,b,e,h \rangle$
5	$\langle a,c,d,e,f,d,c,e,f,c,d,e,h \rangle$
6	$\langle a,c,d,e,g \rangle$
...	...

a = register request,
b = examine thoroughly,
c = examine casually,
d = check ticket,
e = decide,
f = reinitiate request,
g = pay compensation,
and h = reject request

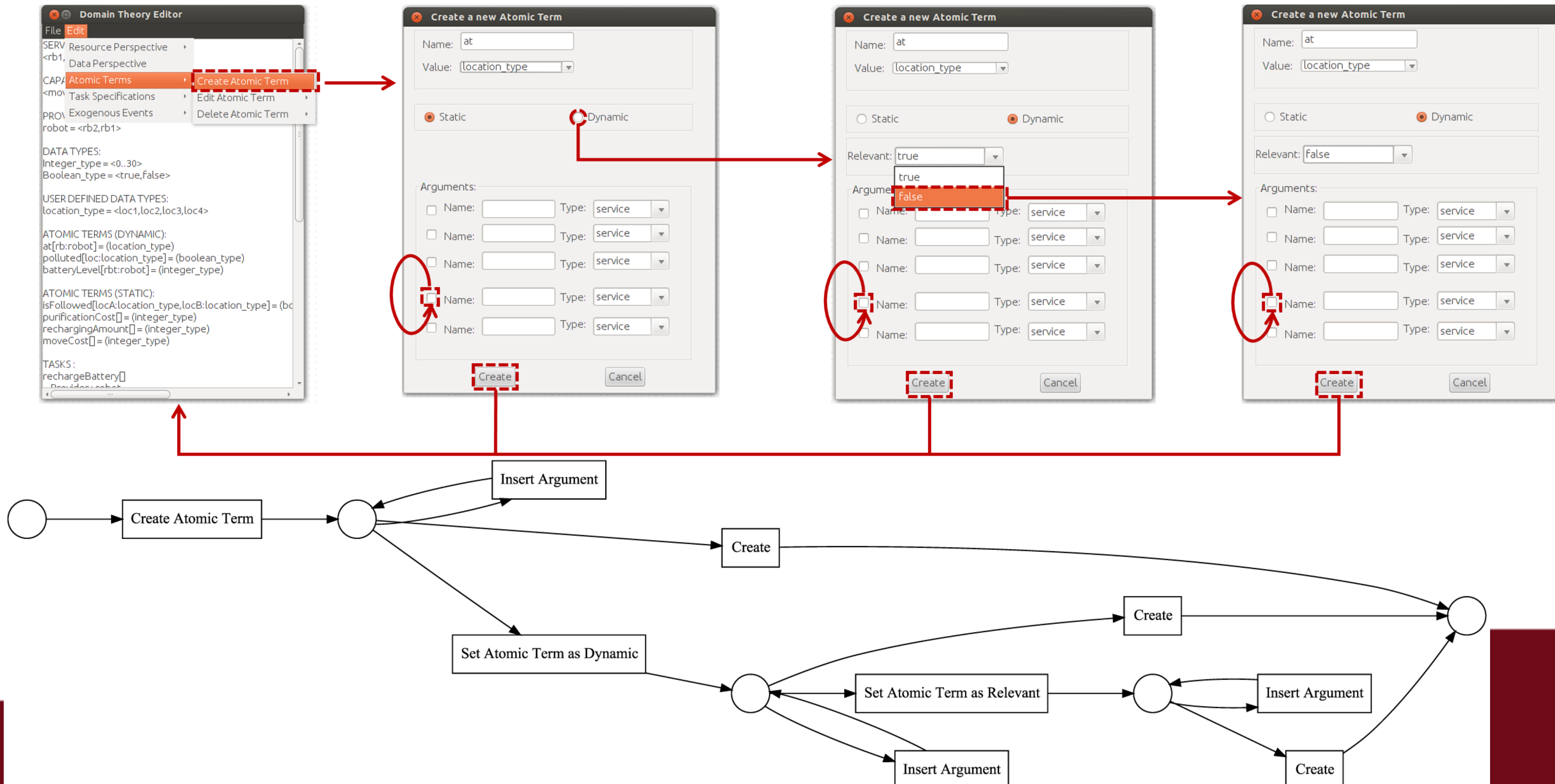
Approach to interpret interaction logs

Given a relevant task to be performed in the UI:

1. Collect interaction logs containing execution traces related to the task.
   - Reflect user's observed behaviour
2. Formalize potential dialog between the user and the UI in form of an interaction model.
   - Represent the expected behaviour to perform a task
3. Construct an **alignment** between any of the traces extracted from the log and the model.
   - Check if the observed behaviour matches the expected one
4. Detect usability issues and in-turn update the UI.

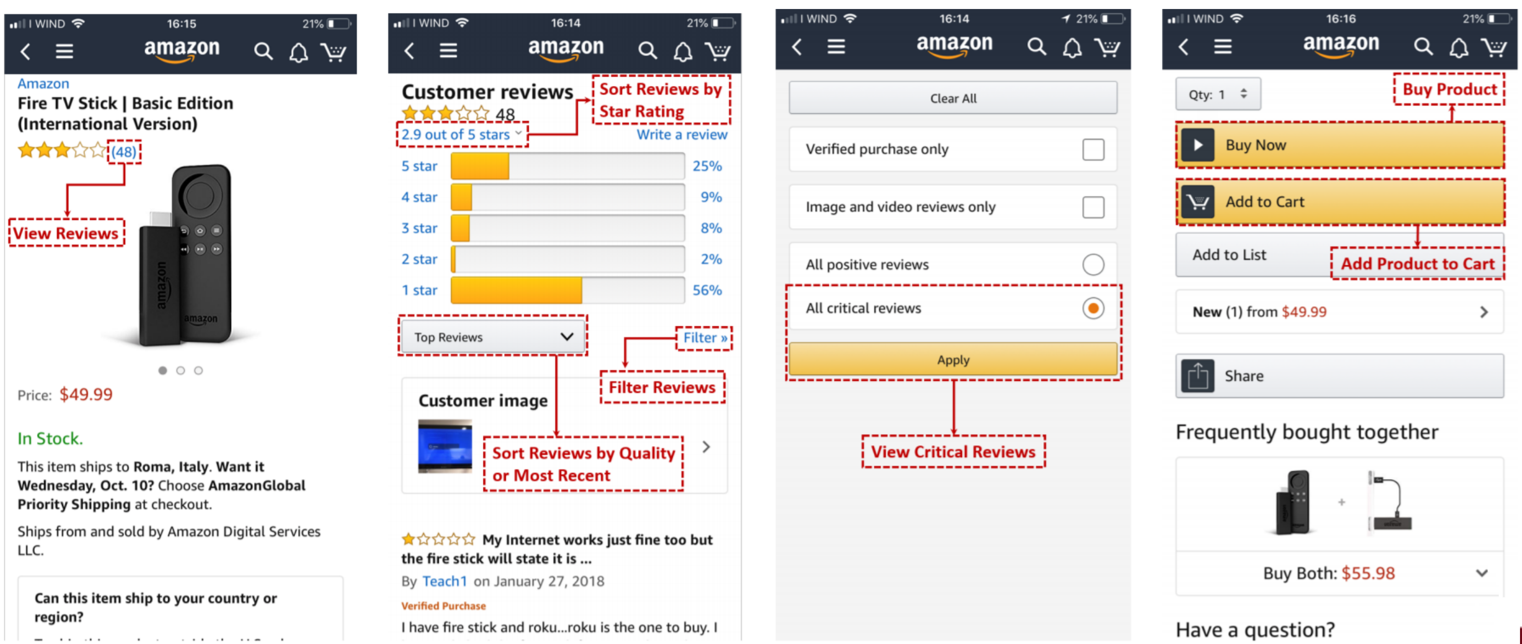
Procedural Interaction Models

- Simple behaviours can be completely captured using procedural models, for example Petri Nets.



Declarative Interaction Models

- More complex behaviours can be captured using declarative languages, for example Declare (grounded on LTL).
  - Models expressed as set of constraints, such that everything that does not violate the model is accepted.



(a) Reading reviews. (b) Filtering reviews. (c) Critical reviews. (d) Adding to cart.

Declarative Interaction Models

If we consider the previous example, we can specify the interaction model that describes the expected behavior underlying the relevant task

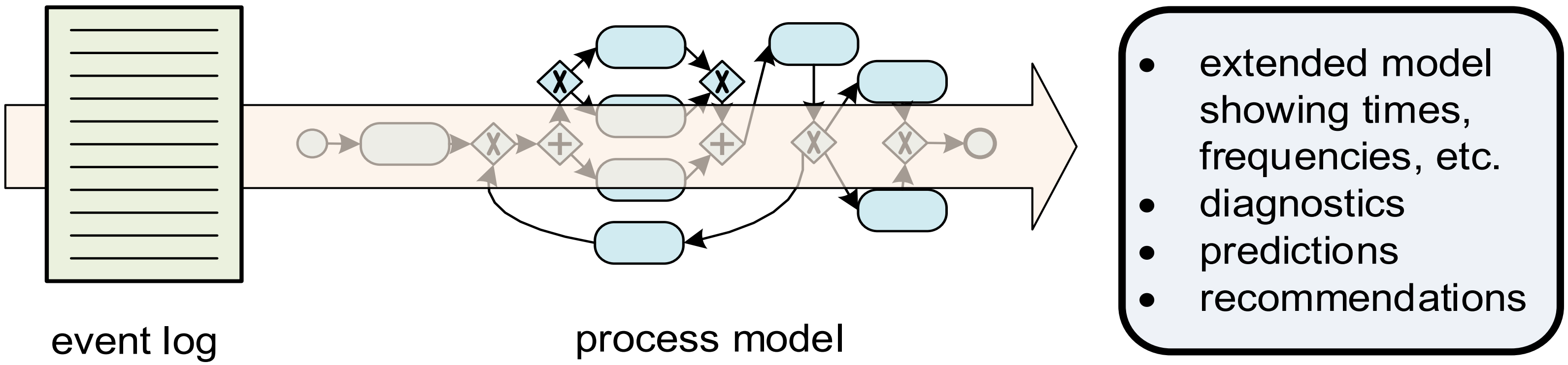
“**read only the critical reviews associated to a Fire TV stick, and then proceeding to buy it**”

as the set consisting of the following declare constraints:

- **absence(s)** means that action s = *Sort Reviews by Quality or Most Recent* cannot ever be performed.
- **existence(b)** means that action b = *Buy Product* must be executed at a certain point of the interaction.
- **precedence(n; b)** forces n = *View Critical Reviews* to precede b = *Buy Product*.

Replay

- Alignment is based on **replaying** log traces on top of the interaction model.
- **Discrepancies** between the log (observed behavior) and the model (expected behavior) can be ***detected*** and ***quantified***.



User mistakes come from bad interaction design

Alignment Example

UPDATED UI

Given that trace T = <v, s, r, b> and based on the interaction model and alignment activity we can identify that:

i. Action s = Sort Reviews by Quality or Most Recent has been executed *even if forbidden*

ii. Action n = View Critical Reviews is required by the model, must be executed before *b* = Buy Product

The alignment of the trace T with the model will be instructed to *skip* action *s* and *insert* action *n* before *b.*

Therefore, the aligned trace is

$$T_3 = \langle v, del(s), r, add(n)\ b \rangle$$

References on Trace Alignment

- **Declarative-based Alignment**
  - Andrea Marrella, Lauren Stacey Ferro, Tiziana Catarci: *An Approach to Identifying What Has Gone Wrong in a User Interaction*. INTERACT (3) 2019: 361-370
- **Petri net-based Alignment**
  - Andrea Marrella, Tiziana Catarci: *Measuring the Learnability of Interactive Systems Using a Petri Net Based Approach*. Conference on Designing Interactive Systems 2018: 1309-1319

Concluding remarks

- Physical-digital
- Smart products
- Cooperative interaction
- <i_w, i_r, s_pr, wash> → <i_w, add (check), del(i_r), s_pr, wash>
- No more (unwanted) pink laundry