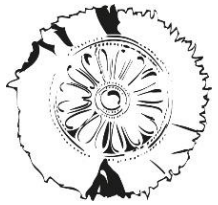
# ReDef: Context-Aware Recognition of Interleaved Activities using OWL 2 and Defeasible Reasoning

Georgios Meditskos, Efstratios Kontopoulos, Ioannis Kompatsiaris

*Information Technologies Institute*
*Center for Research & Technology Hellas*


Information Technologies Institute



CERTH
CENTRE FOR RESEARCH & TECHNOLOGY HELLAS



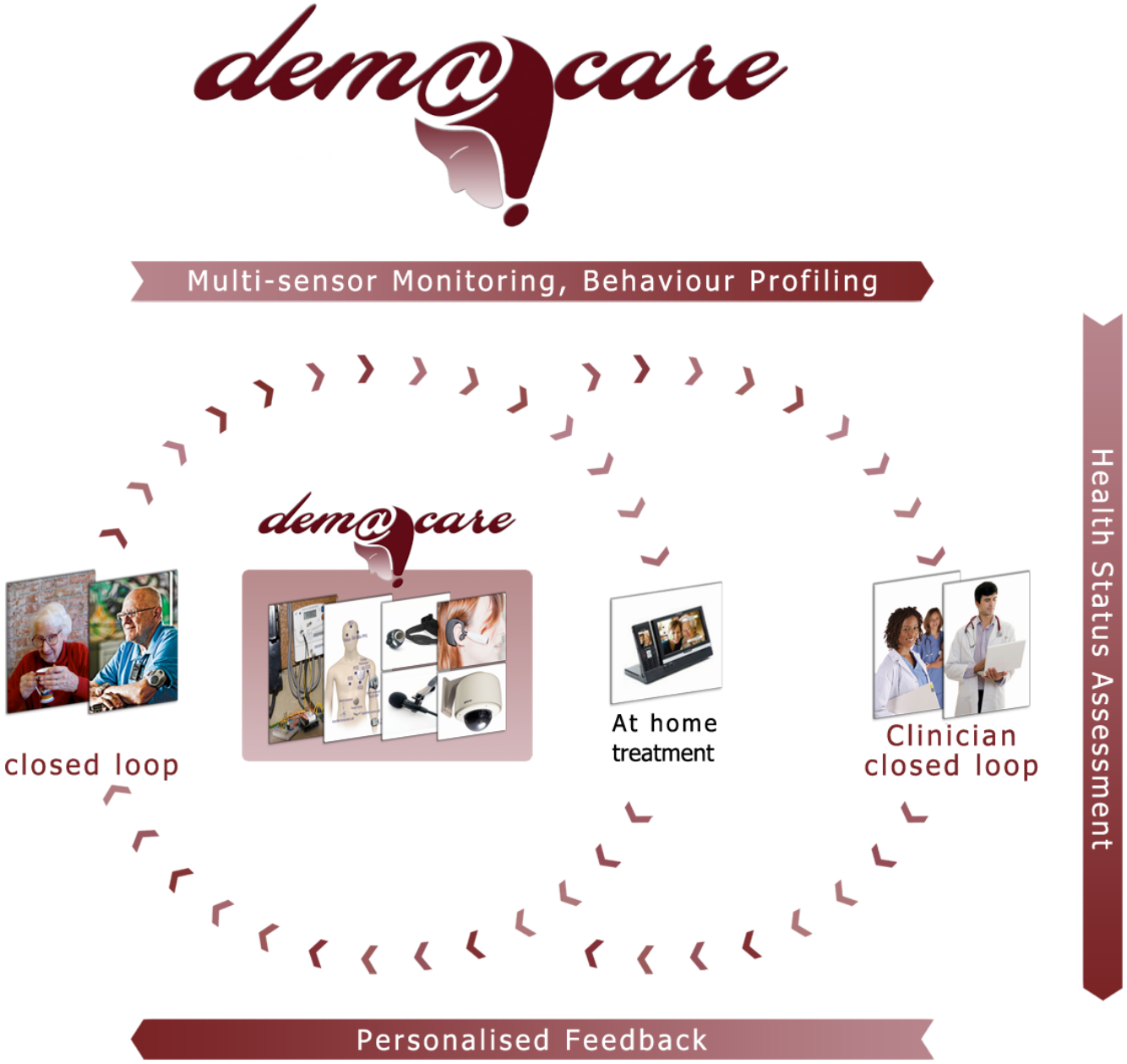
European Commission
Information Society and Media

# Dem@Care Project

- A close-loop multi-parametric remote monitoring framework
  - For timely diagnosis, assessment, maintenance and promotion of self independence of people with dementia
- Enhance clinical workflow by
  - Continuous monitoring the condition and progress of PwD
  - Providing objective multi-sensor measurements

# Dem@Care Project

- Focus on three directions:
  - Implementation of multi-sensor monitoring and analysis of behavior/activities
  - Support person-tailored, time-evolving behaviour profiling & interpretation
  - Support feedback for personalized treatment and care



# Outline

- Introduction
- Knowledge-driven recognition of activities using OWL 2 and SPARQL
- Problem description / Interleaved activities
- The ReDef extension
  - Activity telicity
  - Defeasible rules for handling interleaved activities
- Use case
- Conclusions

# Human Activities in Pervasive Environments

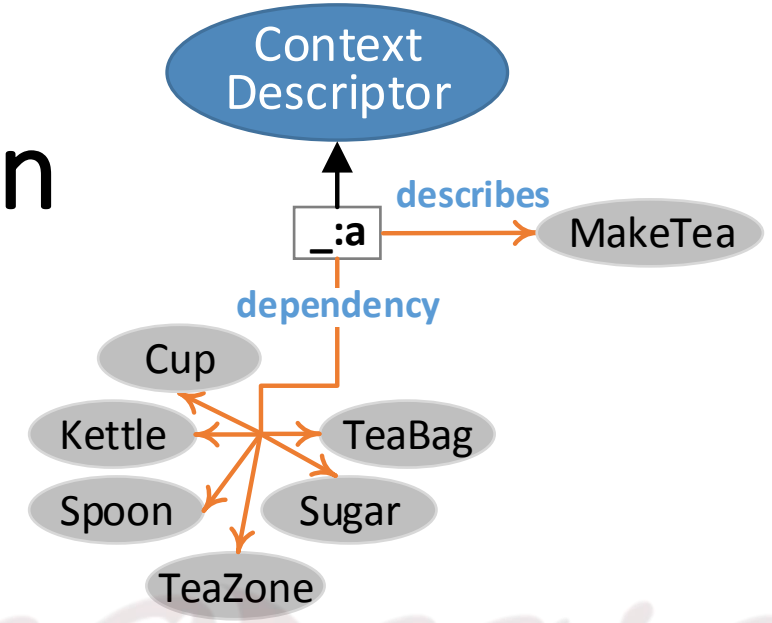
- Smart Homes, Health Care, ...
  - e.g. monitoring the heath status of elderly people
- Key challenges
  - Fusion and correlation of heterogeneous sensors and modalities
    - contact sensors, video analytics, sleep sensors, accelerometers, ...
  - Noise, missing observations, synchronization issues
  - Behavioral variability
    - Different ways the activities are performed (even by the same person)

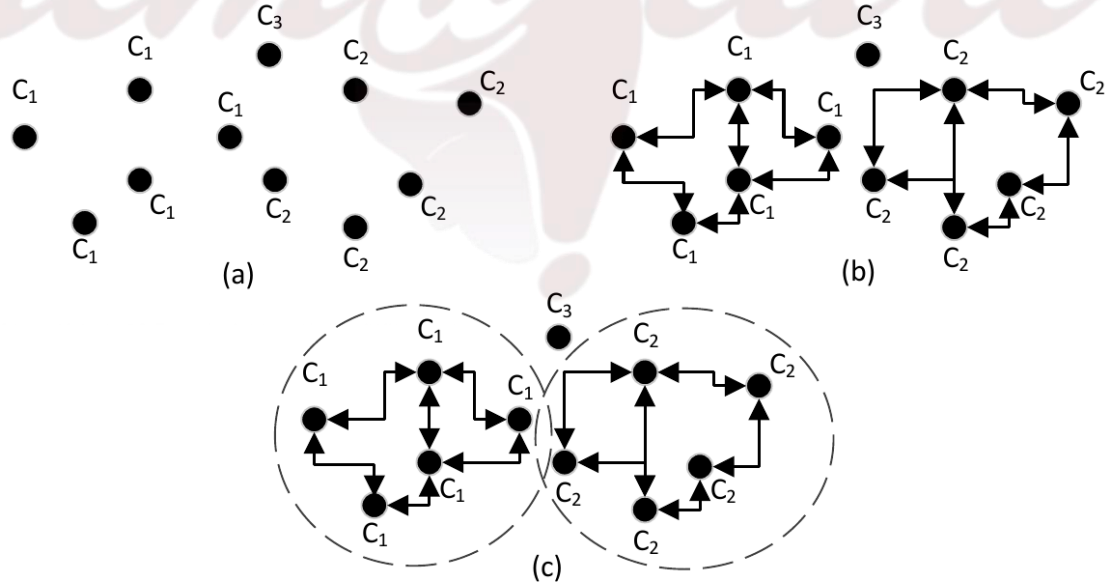
# Ontologies and Activity Recognition

- Knowledge-driven solutions
  - Vocabularies for representing low-level observations
    - objects, locations, events
  - Complex activities
    - e.g. activity hierarchies, activity models
  - Profile/Clinical information
    - habits, trends, abnormal situations / problems, etc.
- Ontologies + Rules
  - Modelling of richer relations
    - e.g. temporal relations
- Ontologies + Data-driven solutions
  - learn activity models, update profile ontologies ...

# Context-Aware Activity Recognition

- Combination of SPARQL and OWL 2 meta-modelling
- Context descriptors
  - dependencies among lower level observations and high-level activities
    - e.g. objects, locations, actions relevant to an activity
- Given a set of low-level observations and a set of situation descriptors, the context-aware algorithm segments the initial trace of observations into meaningful contexts

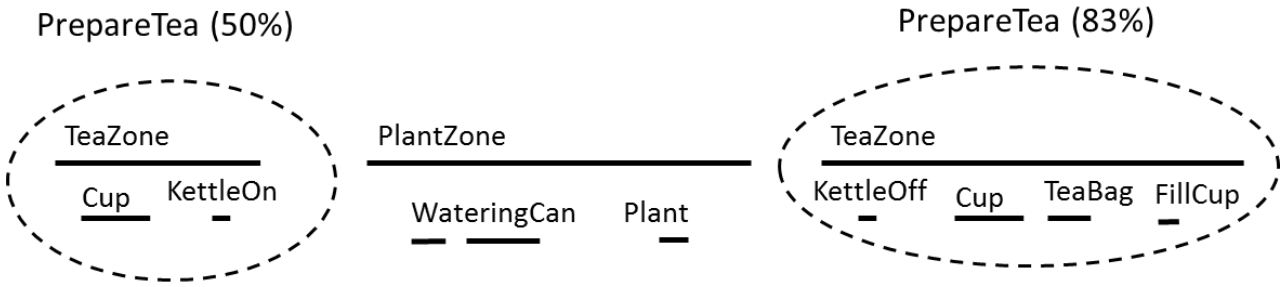



G. Meditskos, E. Kontopoulos, I. Kompatsiaris, Knowledge-Driven Activity Recognition and Segmentation Using Context Connections, ISWC 2014

# Problem/Challenge

- Interleaved activities
  - one activity may be paused in order to perform one or more other activities
- Problem
  - interleaved contexts are recognized as individual activities, affecting the performance
- Challenge
  - classify interrupted instances of the same task as a single activity
- Example
  - Preparing tea and watering the plant




$PrepareTea_{descriptor}$ = {TeaZone, Cup, KettleOn, KettleOff, TeaBag, FillCup}

# Context-Aware Handling of Interleaved Activities

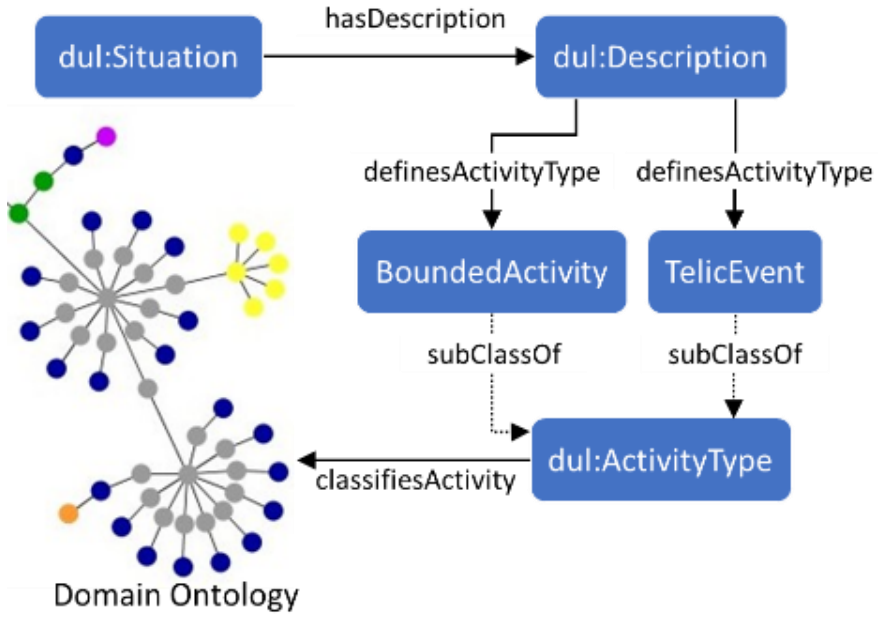
- Trivial solution
  - Time windows: group similar activities within the same time interval
  - Problem: activity duration usually varies
- Our approach
  - Context-aware grouping of interrupted activities by introducing the notion of telicity
    - Groups activities based on the existence of certain observations or certain activity contexts
  - Implementation using Defeasible rules

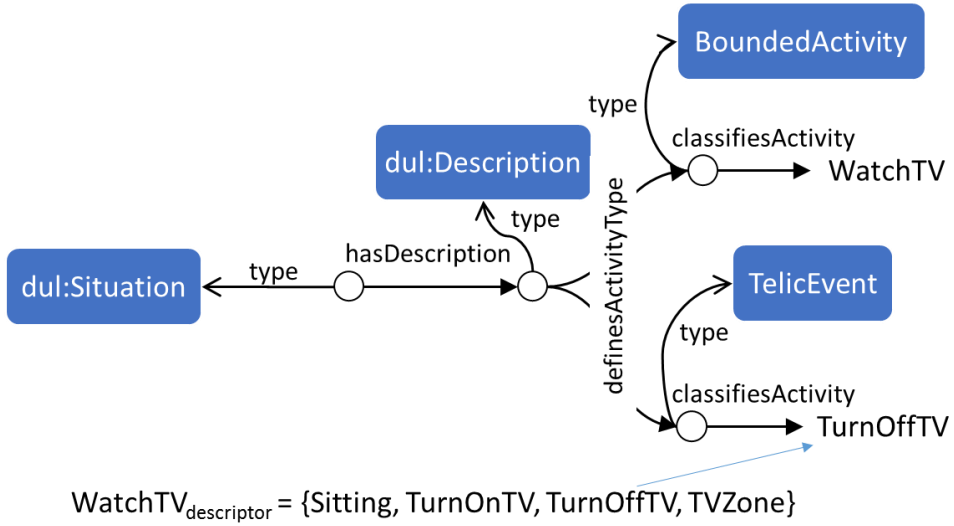
# Activity Telicity: Two types

- *The context that designates when an activity has been completed*
- Two ontology patterns for modelling two types of telicity
  - Telic event
  - Inter-context
- Both ontologies implement the descriptions and situations (DnS) ontology pattern of DOLCE Ultra Lite (DUL)
- Make use of the meta-modelling capabilities of OWL 2 (punning)

# Telic Event Pattern

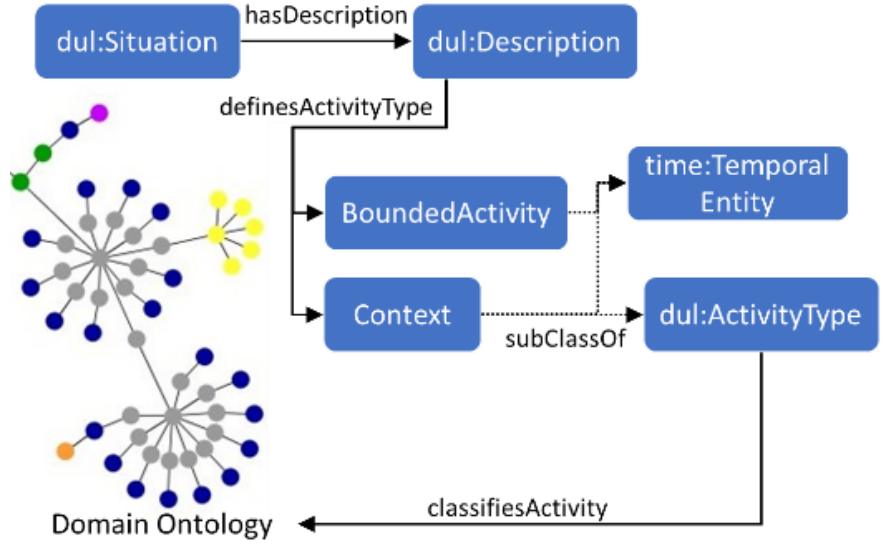
- Defines the terminating state of a complex activity
  - an observation type that belongs to the activity's situation descriptor and denotes the completion of the activity
  - e.g. turning TV off (watching TV)

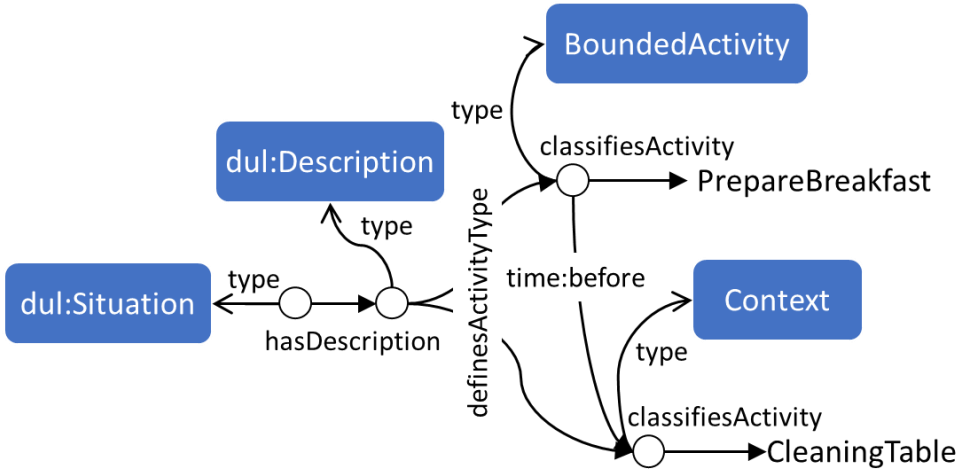


# Inter-context Telicity Pattern

- There are activities that cannot be bounded to specific endpoints
  - e.g. preparing breakfast is a dynamic task that involves many activities without a predefined order or terminating contexts
- Capture activity telicity by means of existence of another context
  - e.g. the detection of an activity relevant to cleaning the table in the morning is an indication that the individual may have prepared a breakfast earlier



# ReDef: Recognising Interleaved Activities

- Given:
  - **Activity traces:** set of detected complex activities with start/end timestamps
  - **Sub-events:** the constituent parts (observations) of the complex activities.
  - **Activity telicity patterns:** instantiations of the patterns
- Examining already detected activities to detect situations when the telicity patterns are satisfied in order to derive interleaved tasks
- Use of Defeasible reasoning to aggregate activities

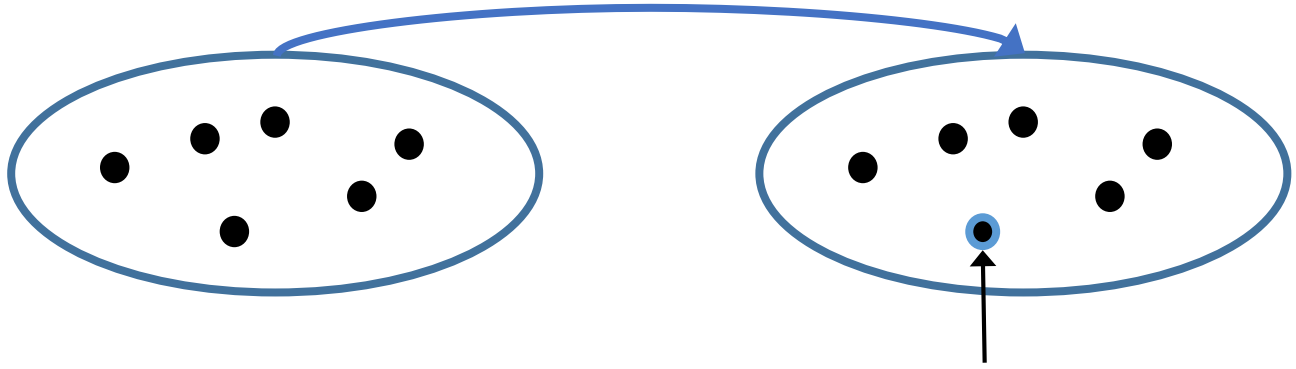
# Defeasible Reasoning

- Non-monotonic logics formalism
  - intuitive knowledge representation
  - conflict resolution mechanisms
- **Strict rules**: A -> p
- **Defeasible rules**: A => p (can be defeated, *holdsFork(X) => havingLunch(X)*)
- **Defeaters**: A ~> p (*sleep*(*X*) ~> ¬*havingLunch*(*X*))
- **Superiority relationship:** for resolving conflicts among defeasible rules
- Some advantages:
  - Low computational complexity
  - Reasoning with incomplete information (critical in sensor environments)
  - More intuitive type of reasoning, much closer to human reasoning especially for the non-accustomed users (e.g. doctors, patients, etc.)

# Telic Event Rule 1

- **$r_1$:** activity(A1,T11,T12), activity(A2,T21,T22), T21 > T12, type(A1,A), type(A2,A), telic(TL,A), subEvent(Z,A2), type(Z,TL) $\Rightarrow$ interleaved(A1,A2)

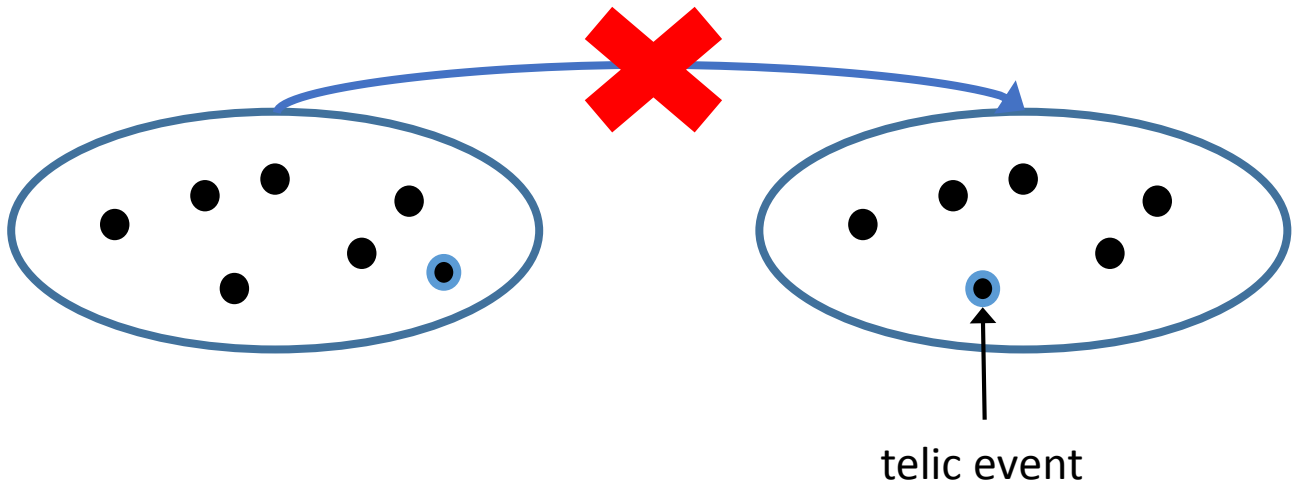
*$r_1$ determines when two separate activities constitute a single, interleaved one, based on the existence of the corresponding telic observation in the activity context that takes place last*



# Telic Event Rule 2

- **$r_2$:** activity(A1,T11,T12), activity(A2,T21,T22), T21 > T12, type(A1,A), type(A2,A), telic(TL,A), subEvent(Z,A1), type(Z,TL) $\Rightarrow$ $\neg$interleaved(A1,A2)

*$r_2$ establishes an exception to $r_1$ that takes place when the first activity (also) includes a telic observation.*

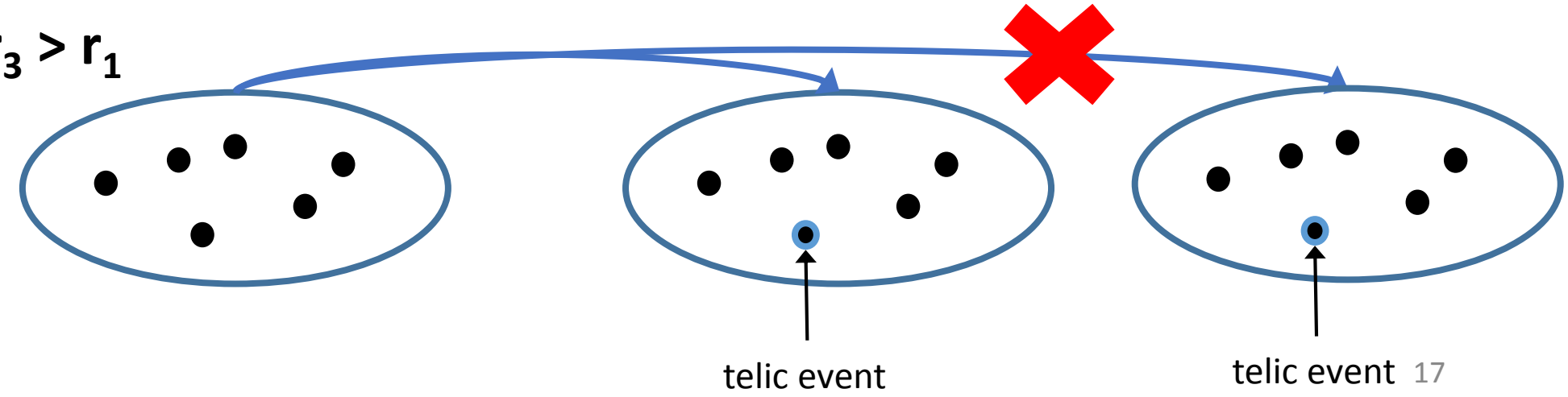


# Telic Event Rule 3

- **$r_3$:** activity(A1,T11,T12), activity(A2,T21,T22), activity(A3,T31,T32), T21 > T12, T31 > T22, type(A1,A), type(A2,A), type(A3,A), telic(TL,A), subEvent(Z1,A2), subEvent(Z2,A3), type(Z1,TL), type(Z2,TL) $\Rightarrow \neg$interleaved(A1,A3)

*$r_3$ ensures that an activity is linked only with the most recent telic context*

- **$r_2$, $r_3$ > $r_1$**

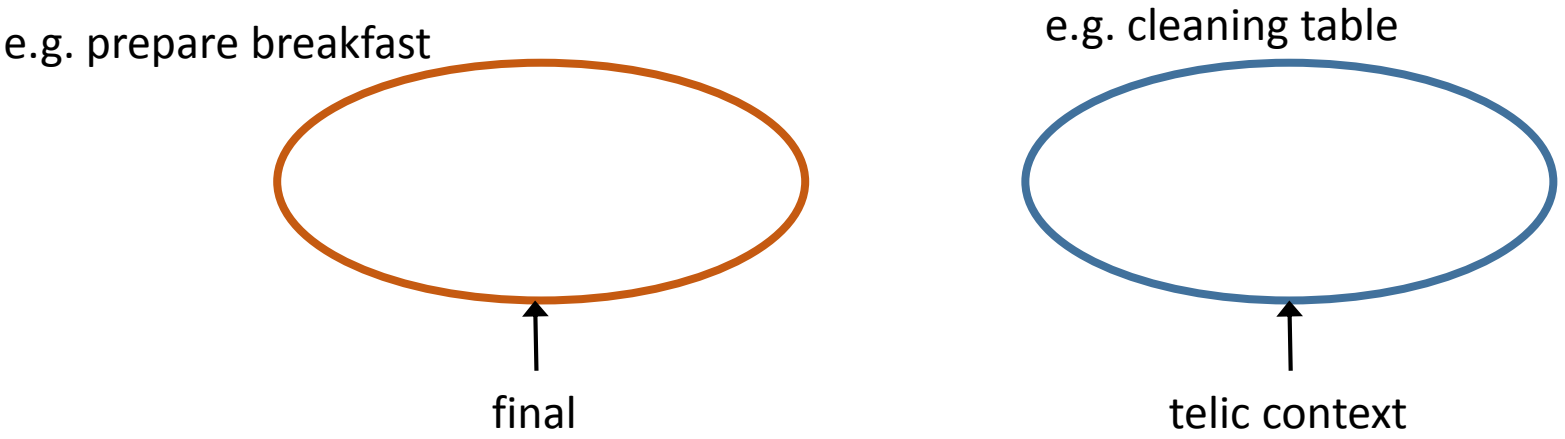


# Inter-Context Telicity rules

- **$r_4$:** activity(A1,T11,T12), activity(B1,T21,T22), latest(A1,B1), type(A1,A), type(B1,B), telicContext(A,B) $\Rightarrow$ final(A1)

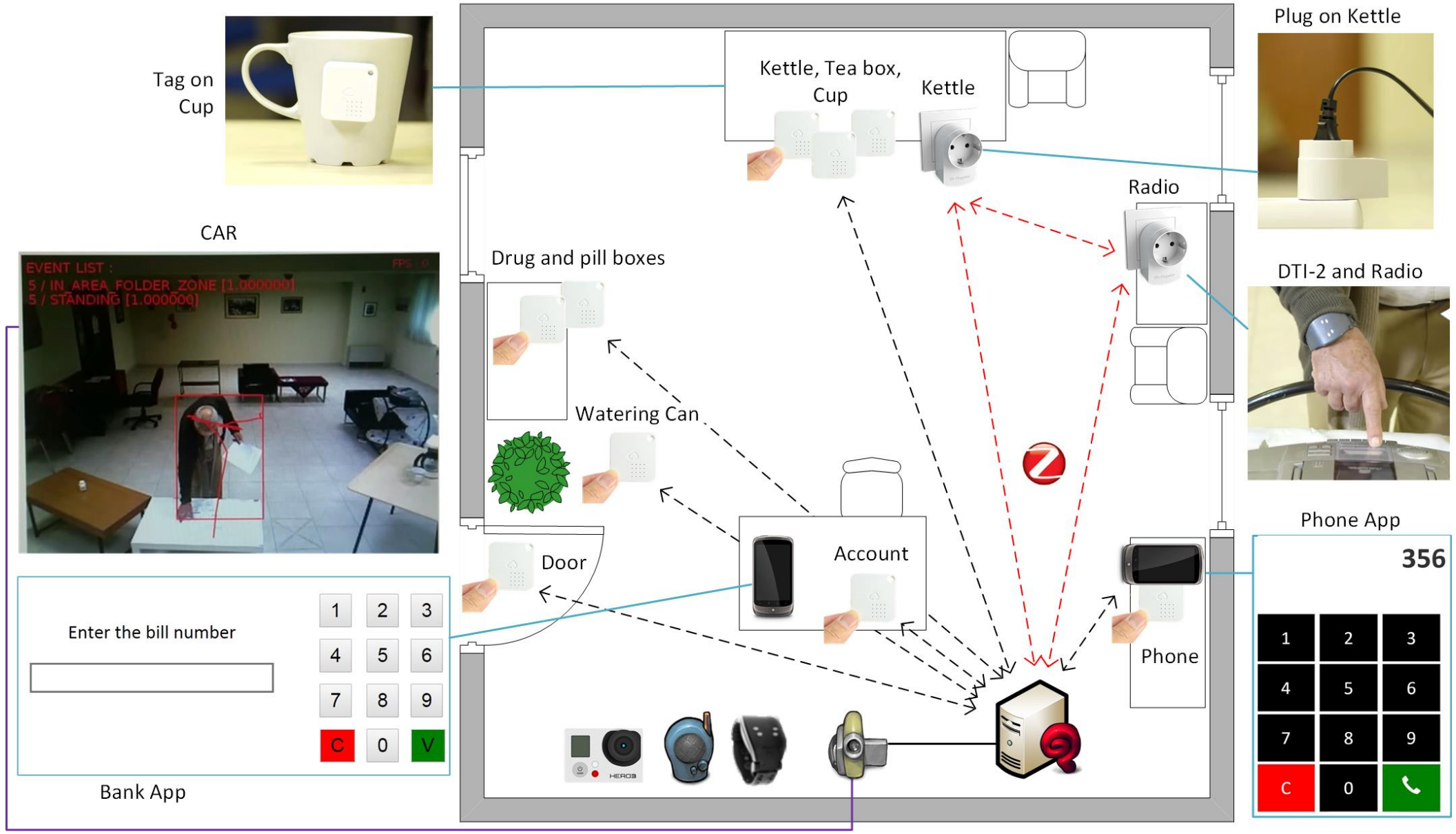
`final(A):` indicates that activity A is completed (no subsequent activities of the same type may be appended to A)

`[latest(A1,B1), type(A1,A), type(B1, B)]` retrieves the closest most recent activity of type A to type B



# Use Case

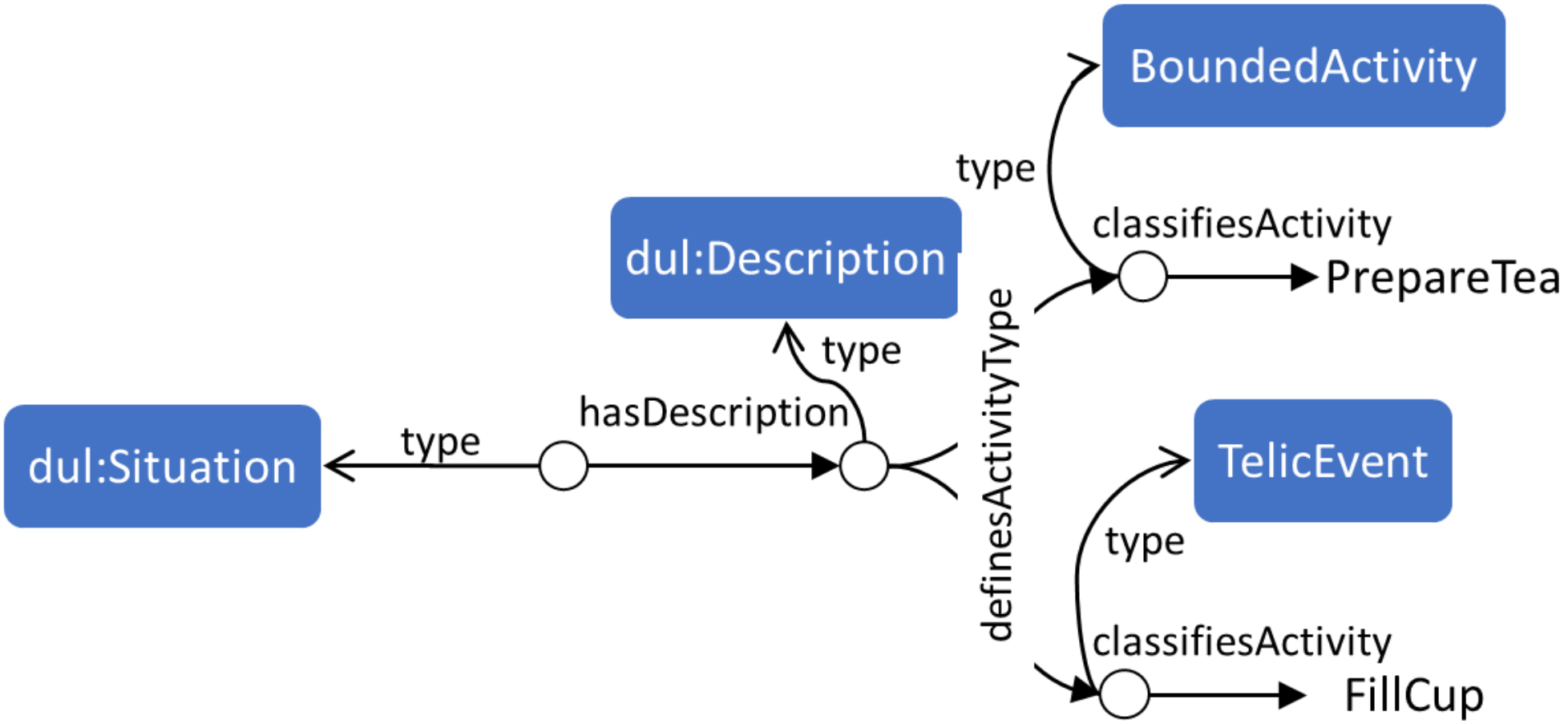
- ReDef is part of an Activity of Daily Living (ADL) recognition framework deployed in a hospital for monitoring Alzheimer's disease patients
- The aim of this deployment is to help clinicians assess the condition of individuals, based on a goal-directed protocol.
  - preparing the drug box, talking on phone, preparing tea and watering the plant.


Tag on Cup
Kettle, Tea box, Cup
Kettle
Plug on Kettle
Radio
CAR
EVENT LIST :
5 / IN_AREA_FOLDER_ZONE [1.000000]
5 / STANDING [1.000000]
Drug and pill boxes
DTI-2 and Radio
Watering Can
Door
Account
Phone App
356
Enter the bill number
1 2 3
4 5 6
7 8 9
C 0 V
Phone
1 2 3
4 5 6
7 8 9
C 0
Bank App

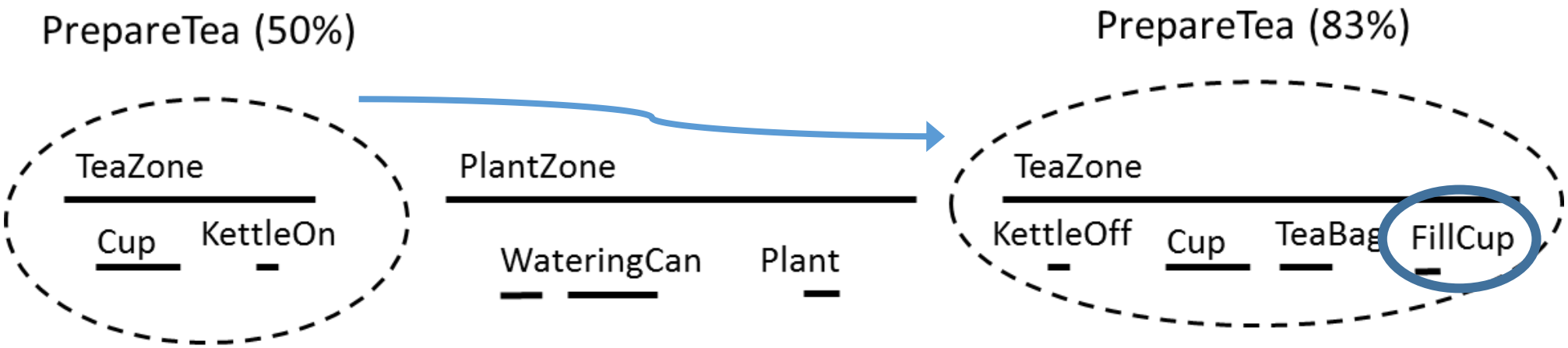
# Use Case

- The majority of the tasks involved in the protocol can be performed in a sequential manner
- low accuracy in detecting the preparation of hot tea performed in an interleaved manner
  - after putting water in the kettle and turning the kettle on, participants went on with other tasks

# Usage of Telic Event Pattern



# Grouping of Interrupted Contexts




$PrepareTea_{descriptor} = \{TeaZone, Cup, KettleOn, KettleOff, TeaBag, FillCup\}$

# Evaluation

- Preliminary results are very promising
  - Testing so far with a small number of participants – ongoing pilots
- Problems have been identified in cases when the analysis modules fail to detect the telic event of an activity, e.g. FillCup in our example
  - In this case telicity cannot be inferred and the detection of interleaved activities fails.
- We are currently investigating the extension of the defeasible rules so as to handle missing information, e.g. by integrating negation-as-failure or more refined/explicit rules expressing exceptions.

# Summary

- ReDef: framework for handling interleaved activates based on contextual information
- OWL 2 ontologies to capture the notion of telicity
  - the context that designates the end of an activity
  - Two types of telicity
    - Telic event
    - Inter-context telicity
- Defeasible rules implement the semantics of the two telicity types
- Practical use case on the healthcare domain

# Future Work

- Identify/Define more types of telicity
- Handle missing information / sensor misinterpretations
- Deploy the framework in home settings
- Extensively evaluate the performance

# Thank you!