

Assumption-based Risk Identification Method (ARM) in Dynamic Service Provisioning

The risk of unstated assumptions

Alireza (Shahin) Zarghami

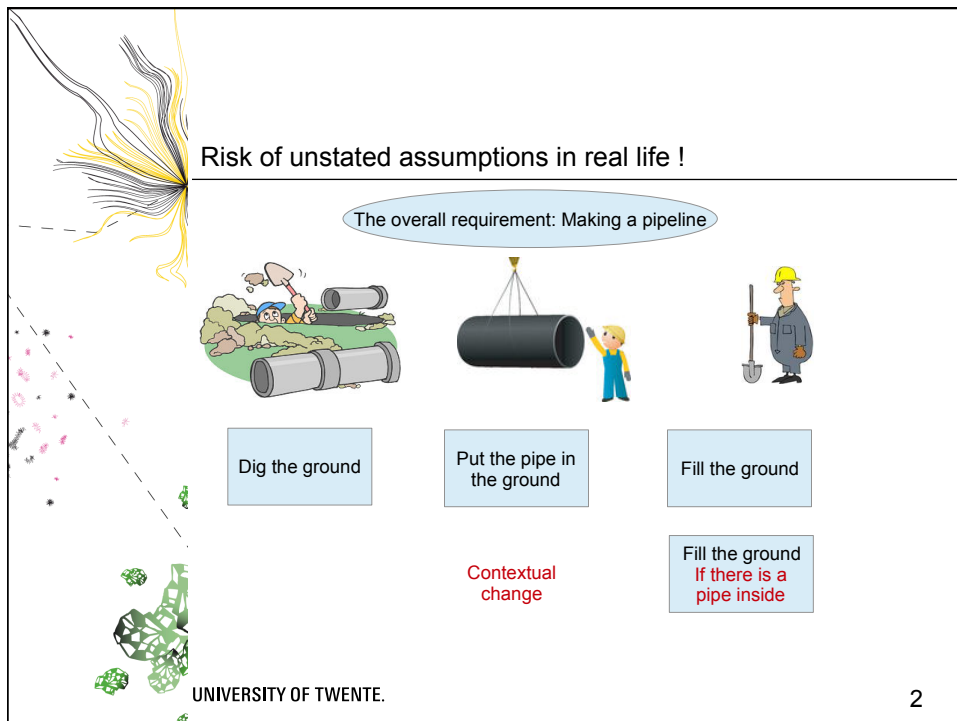
Eelco Vriezekolk

Mohammad Zarifi

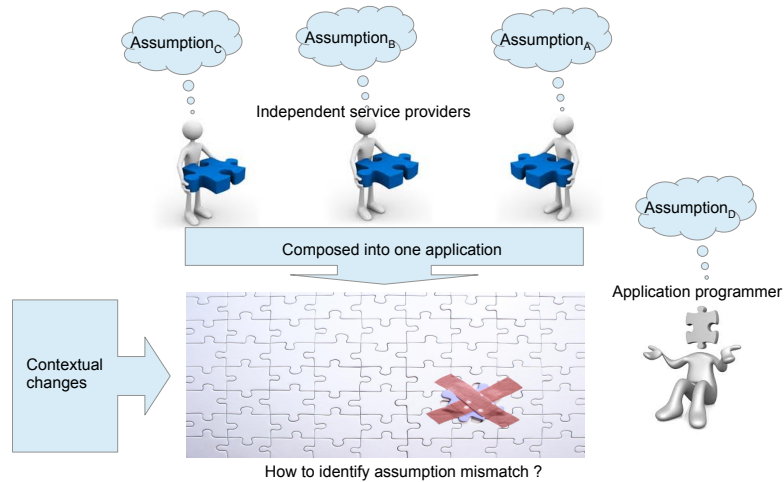
Marten van Sinderen

Roel Wieringa

RE, Rio, July 2013



Risk of unstated assumptions in a dynamic service provisioning ?



UNIVERSITY OF TWENTE.

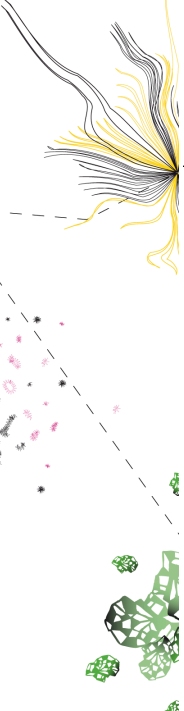
3

The problem

- Each of the contributing service providers makes **assumptions**. if these assumptions were correct, the provided component services would satisfy the requirements on the composite application.
- Each service provider has its **own requirement** which might be **unknown** to others.
- In practice, these assumptions are mostly **unstated**, and some of them are **incorrect**, or at least **mutually inconsistent** across different service providers.
- A composite application is subject to several **foreseen/unforeseen contextual changes** at runtime.
- The unstated assumptions could cause **unexpected behavior** of the application that could lead to a risk

UNIVERSITY OF TWENTE.

4



We are the first one who identified this problem!


- D. Garlan et al., "Architectural mismatch: Why reuse is so hard", 1995
- HAZards and OPerability studies (HAZOP), 1997
- STPA hazard analysis method, 2010
- ...

But we emphasize on **dynamic service provisioning**:

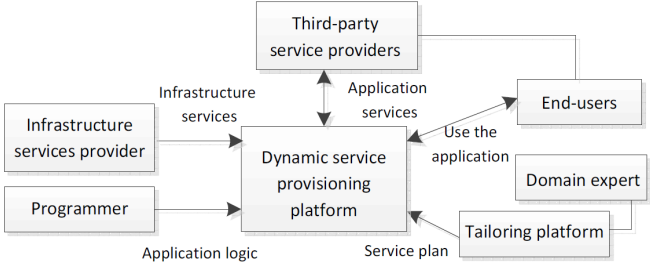
- Individual stakeholders
- No dedicated services
- Several actors are responsible for the entire system design
- Facing foreseen/unforeseen changes

UNIVERSITY OF TWENTE.

5



A generic dynamic service provisioning platform



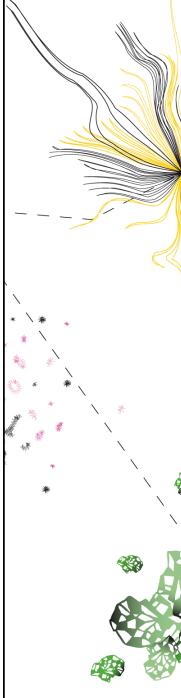
```

graph LR
    subgraph Left
        ISP[Infrastructure services provider]
        P[Programmer]
    end
    subgraph Top
        TSP[Third-party service providers]
    end
    subgraph Right
        E[End-users]
        DE[Domain expert]
        TP[Tailoring platform]
    end
    DSCP[Dynamic service provisioning platform]

    ISP -- "Infrastructure services" --> DSCP
    P -- "Application logic" --> DSCP
    TSP <--> |"Application services"| DSCP
    DSCP -- "Service plan" --> TP
    TP --> DE
    DE -- "Use the application" --> E
    E --> TSP
  
```

UNIVERSITY OF TWENTE.

6



U-Care project

- U-Care project: Develop a **service layer** for **integrated** homecare systems, which provides **tailorable**, **evolvable** and **non-intrusive** homecare services

Applications

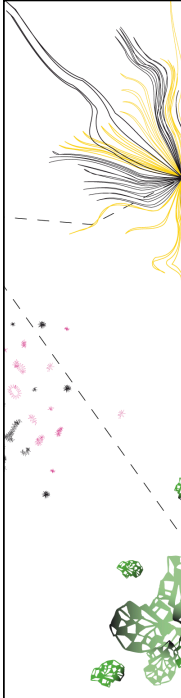
Integration <

Tailoring

Business and architecture

UNIVERSITY OF TWENTE.

7



The overall architecture of U-Care platform

Service plan

Tailoring platform

Care-giver

Professional /Social

Provisioning platform

Tailoring interface

Infrastructure services

Application services

Controlled services


Third-party service

Care-receiver

Zarghami, A., Zarifi Eslami, M., Sapkota, B., van Sinderen, M. Dynamic Homecare Service Provisioning Architecture. IEEE International conference on Service-Oriented Computing and Application, SOCA11, Irvine, USA, December 2011.

UNIVERSITY OF TWENTE.

8




Our homecare field experiment


Three types of applications:

- 1) Vital-sign monitoring
- 2) Medication monitoring
- 3) Social activity

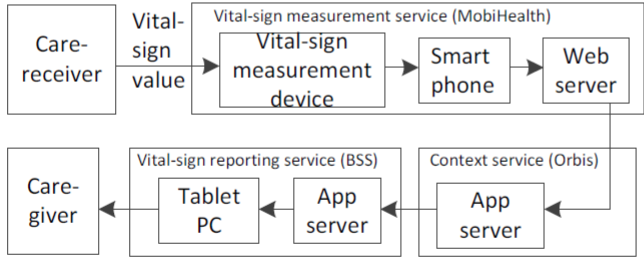
- 3 industry service providers
- 2 academia service providers
- 8 care-receivers, 4 care-givers
- 2 field experiments
- 400,000 transactions in total



9



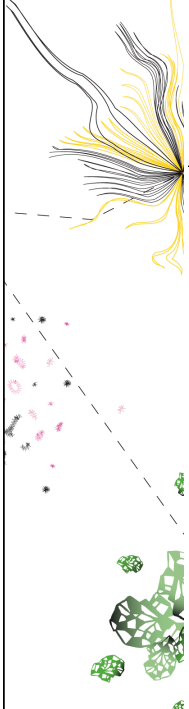
Vital-sign monitoring application



Contribution argument:

If service providers S_1, \dots, S_n behave like this: A_1, \dots, A_n , respectively, then the composite application satisfies its requirements.

10



Vital-sign monitoring application

Since all the service providers followed the provided service specifications we were sure that the system will work ☺

We faced several problems ! ☹

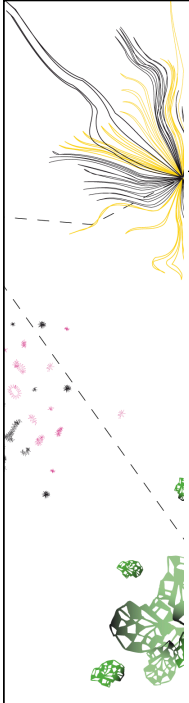
Three types of problems:

1. Service availability problem
2. Data transportation problem
3. Data storage problem

These problems cause several risks

UNIVERSITY OF TWENTE.

11



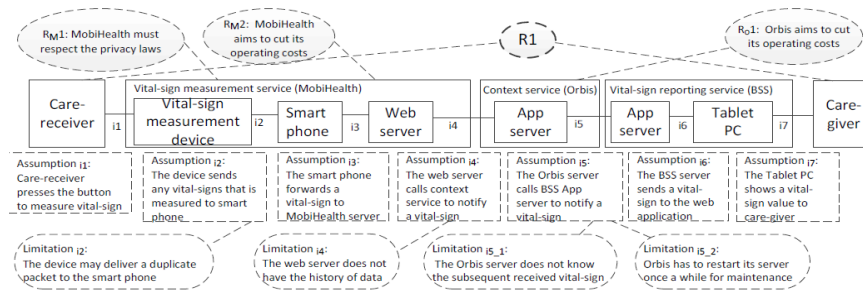
Assumption-based Risk Identification Method (ARM)

- 1) Translate the main requirements into assumptions on the interfaces of the network actors (based on contribution argument)
- 2) Explore the capabilities and limitations of each service provider to satisfy the assumptions on its interface.
 - Service availability questions
 - Data transportation questions
 - Data storage questions.
- 3) Explore in which way each limitation identified in the previous step could cause the service provider fail to meet an assumption.

UNIVERSITY OF TWENTE.

12

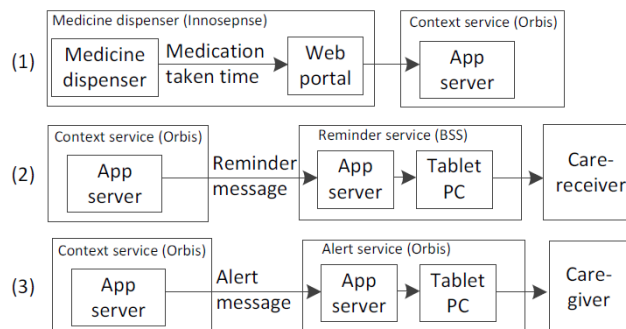
Applying the ARM method on vital-sign monitoring application



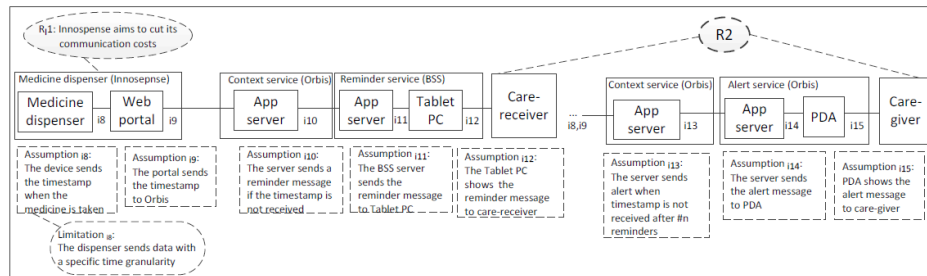
R1: The care-receiver shall provide vital-sign values to the care-giver according to their treatment plan

Applying the ARM method on medication monitoring application

The application consists of three networks



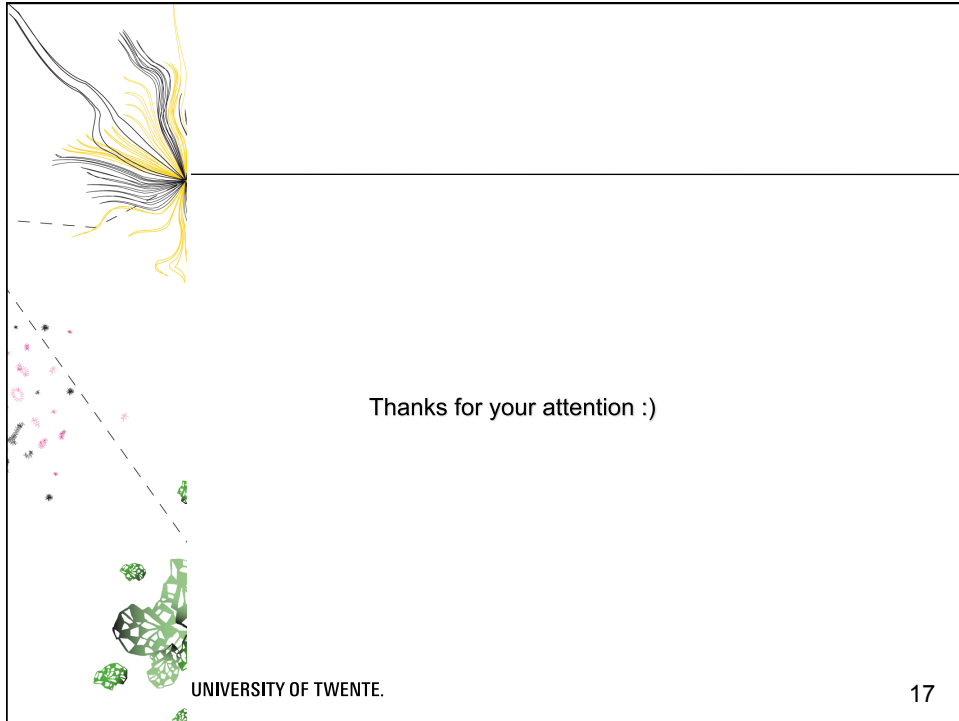
Applying the ARM method on medication monitoring application



R1: The care-receiver shall take a medicine from the dispenser at the scheduled time according to his service plan.

Conclusion & future work

- Unstated assumptions could cause several risks in dynamic service provisioning
- To find the assumption, to some extent, we should know the internal implementations of service providers (service autonomy principle?)
- To solve the problem, one/several actors should pay the cost and to some degree compromise their requirements. (How to negotiate?)
- Translating the requirement to the assumption needs to be done in more formal way
- Predicting changes in end-user behavior after introducing a system as what if scenarios is not a straightforward task at the design time



Thanks for your attention :)