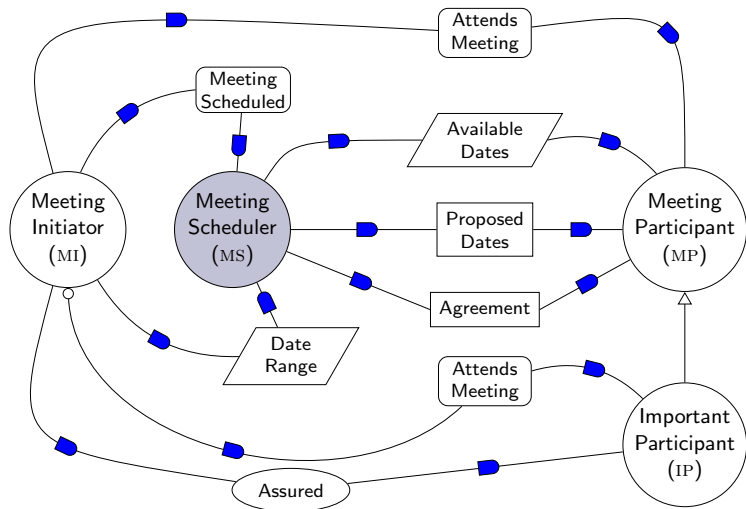


Goal Modeling Concepts in Tropos

NB: Different meaning of the word “actor” from actor programming

- ▶ Actors: stakeholder or software “system”
- ▶ Goals
 - ▶ “Hard” by default: required functionality
 - ▶ “Soft” goals: support partial fulfillment
- ▶ Decomposition: AND or OR
- ▶ Beliefs; Plans; Resources
- ▶ Dependencies between actors
 - ▶ In terms of their goals, plans, or resources
- ▶ Contributions of any goal or plan to a soft goal
 - ▶ Makes (++); Helps (+); Hurts (-); Breaks (--)
- ▶ Spans requirements, design, specification, and implementation phases

Meeting Scheduler: Traditional View



Simplified Goal Modeling

The “system” actor corresponds to the application being built

- ▶ Identify actors: stakeholders plus one or more “system” actors—both “as is” and “to be” (placeholders)
- ▶ Elicit goals of each stakeholder actor
- ▶ Decompose such goals based on domain knowledge
 - ▶ Available services for traditional services
 - ▶ Applicable context abstractions (beliefs) for context-aware apps
 - ▶ Beliefs about context abstractions affect choice of goal at run time
- ▶ Identify contribution links to soft goals
- ▶ Identify dependencies between actors
- ▶ Incrementally, assign goals to system actors
- ▶ When all stakeholders goals are supported by system actors’ goals, design and implement system actors

Engineering Context-Aware Applications

Joint work with Pradeep Murukannaiah

How can a developer capture a user's mental model of contexts?

Traditional context-aware computing

- ▶ Treats context objectively: fixed across applications and users
- ▶ Disregards subjective aspects: user expectations, i.e.,
 - ▶ *Why* does a user want something to be done at a certain context?

Traditional goal modeling

- ▶ Captures a user's (agent's) expectations, but doesn't tie those expectations to contexts
 - ▶ Provides no guidance for dealing with the environment

Xipho: Context-Aware Personal Agents (CPAs)

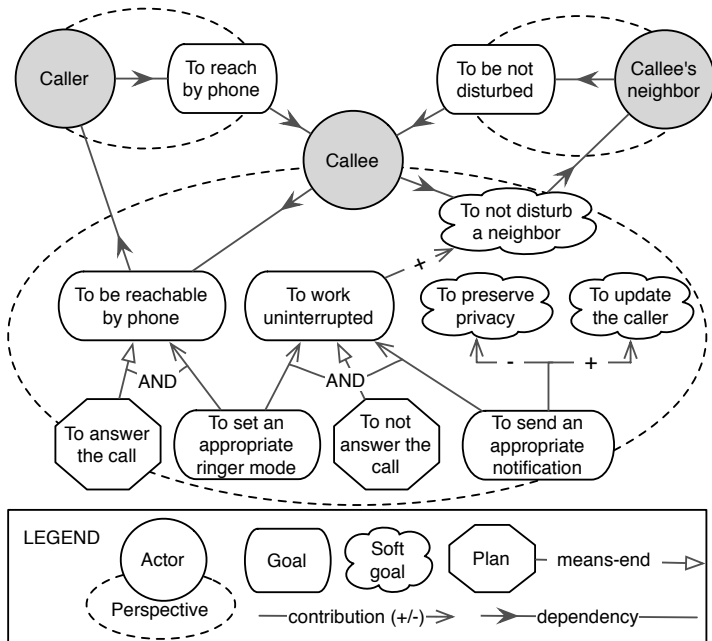
Overview

- ▶ Treats context as a cognitive notion throughout development
- ▶ Captures a user's mental model of contexts
- ▶ Relates contexts to cognitive notions such as a user's goals and plans
- ▶ Guides a developer to make user's context-based expectations explicit

Benefits

- ▶ Reduces time and effort required to develop a CPA
- ▶ Yields CPA designs that are easy to comprehend for a developer

Example: Ringer Manager Agent (RMA) for Cell Phone



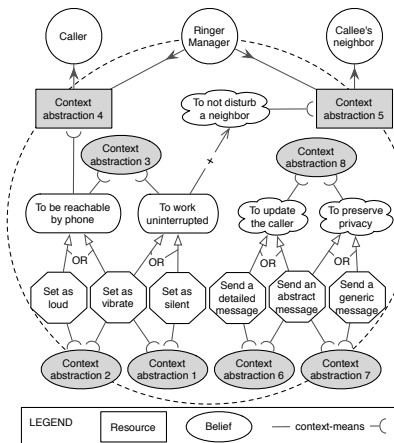
Step 1: Context-Means Analysis

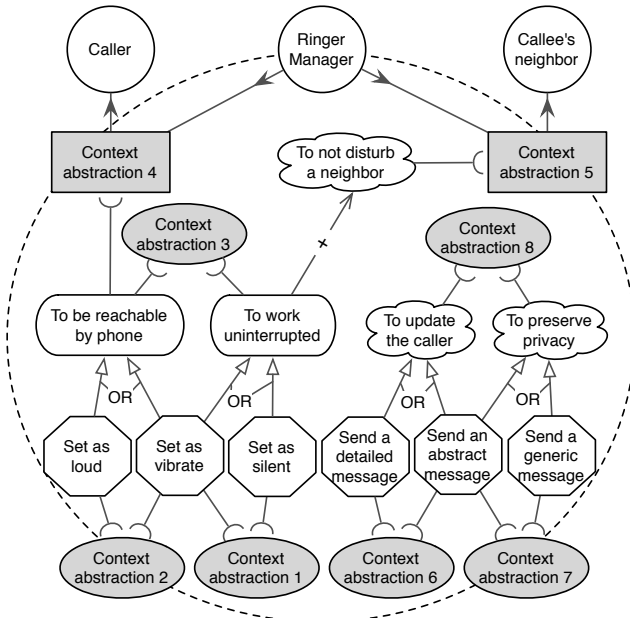
Determining where context can help select between goals and plans

Identify scenarios in which context is important

Analyze

- ▶ Conflicting goals
- ▶ OR decompositions
- ▶ Soft goals
- ▶ Dependencies



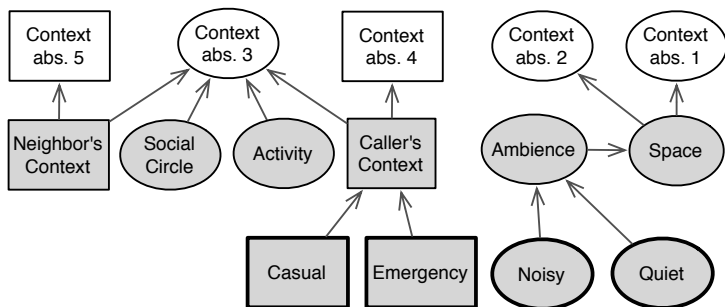


Step 2: Context Information Modeling

Mapping elements of context to information sources

Tailor a generic context metamodel to each identified scenario

RMA-Specific Context Model



Step 3: Contextual Capability Modeling

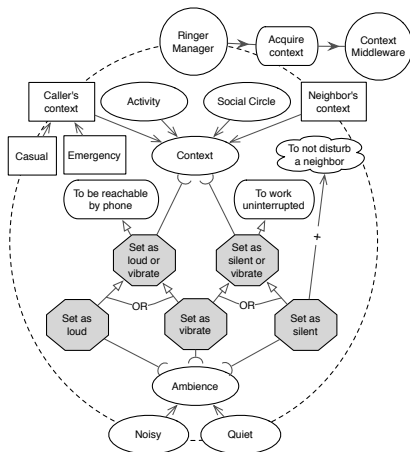
Mapping contexts to decisions

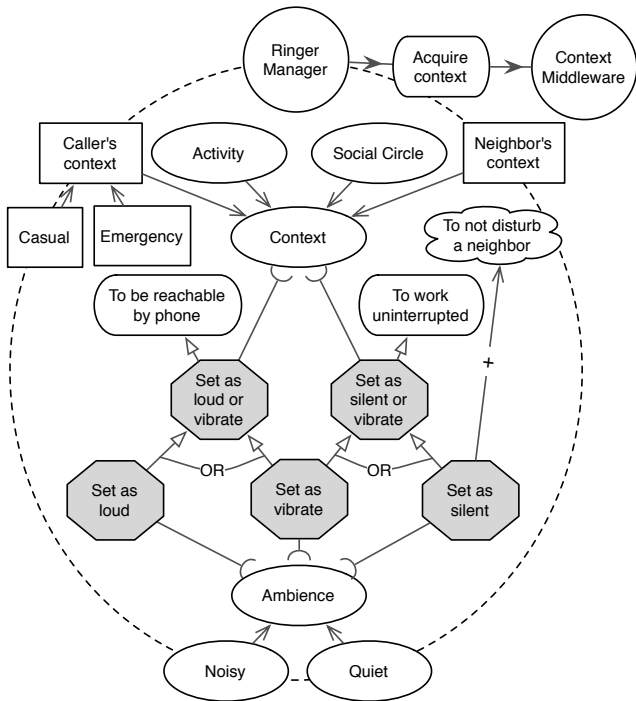
Specify a set of rules that map context instances to agent capabilities

An example rule

$(Activity = ?A_1) \wedge$
 $(Social\ circle = ?S_1) \wedge$
 $(Neighbor's\ context = ?N_1) \wedge$
 $(Caller's\ context = Emergency)$

→ *Set as loud* or *Set as vibrate*





Step 4: Context Middleware Architecture

Enabling computation of context via universal middleware

Map tasks common to all CPAs to a middleware agent

The middleware

- ▶ Elicits context instances
- ▶ Recognizes context instances
- ▶ Acquires context resources

