

# Modeling and Analyzing Variability for Mobile Information Systems

Raian Ali, Fabiano Dalpiaz, and Paolo Giorgini

University of Trento - DISI, 38100, Povo, Trento, Italy.  
{raian.ali, fabiano.dalpiaz, paolo.giorgini}@disi.unitn.it

**Abstract.** Advances in size, power, and ubiquity of computing, sensors, and communication technology made possible the development of mobile or nomadic information systems. Variability of location and system behavior is a central issue in mobile information systems, where behavior of software has to change and re-adapt to the different location settings. This paper concerns modeling and analysis of the complementary relation between software and location variability. We use graphical and formal location modeling techniques, show how to elicit and use location model in conjunction with Tropos goal-oriented framework, and introduce automated analysis on the location-based models.

## 1 Introduction

Advances in computing, sensing and communication technology have recently led to the growth of interest in software mobility. Mobility emphasizes several concerns (space, time, personality, society, environment, and so on) often not considered by traditional desktop systems [1, 2]. Besides computing ubiquity, the 21st century computing [3] is expected to have a core “mental” part: computing systems act on behalf of humans executing tasks without prompting them for and receiving their explicit requests, i.e. computing will realize the concept of agency, and interaction with computers is getting hence as minimal as possible [4]. Advances in technology do not necessarily imply the easiness of exploiting it, rather more challenges are introduced. Software systems can be given more responsibility, and they can now actively support several decision making processes. Consequently, appropriate software development methods and models need to be developed, or adapted, to cope with the new software scenarios.

Nomadic users expect smarter information systems, able to adapt their behavior without human intervention. Systems should be able to reason about the surrounding location, which includes users themselves, and adapt autonomously their behavior to location settings. What we need then is to model and analyze the variable locations the users can be part of, and define how location influences the behavior of the mobile information system (hereafter MobIS). Changing behavior needs a kind of reasoning on what exists and what can be done based on criteria like user preferences, cost, time, priority, and so on. Besides modeling the variant locations, and the different behaviors the MobIS can adopt accordingly, we need to define the process of analyzing and designing such systems that are distinguishably different from desktop driven information systems.

In this paper, we use Tropos [5, 6] goal-oriented framework to analyze and model the different behaviors a MobIS can adopt. We use goal model to represent the alternative strategies for satisfying a goal, and then we attach location properties to a set of defined variability points on goal model. The integration between goal model and the location where the software operate, will allow us to support the decision making process when deriving a location-tailored MobIS instance as well as enabling different kinds of analysis. MobIS different strategies for satisfying a stakeholder goal, and location model are designed graphically by designers, and then formalized as a basic step for automated reasoning. Automated reasoning will allow to answer questions like: “*are all MobIS objectives achievable in a given location?*”, “*what is the optimal alternative to achieve an objective in a given location?*”, and “*what is the optimal set of modifications that have to be done in a given location to satisfy some MobIS objectives?*”.

The paper is structured as follow: in Section 2, we review and discuss related work. In Section 3, we present a MobIS motivating example. In Section 4, we discuss how Tropos goal model can be adopted to analyze and represent the relation between goal satisfaction alternatives and location; we also use Tropos conceptualization to give a location definition. In Section 5, we show location-based goal modeling. In Section 6, we model location and location properties captured by the location-based goal model. In Section 7, we describe three kinds of automated analysis, and in Section 8 we draw conclusions and future work.

## 2 Related Work

Mobility is considered as the most important market and technological trend within ICT. Spatial mobility is about people moving around, having wireless access to information and services, but much more other factors imply mobility, like for instance the users’s social, personal and task context [1]. Compared to traditional desktop driven information systems, MobIS has two main differences: limitation of hardware (energy, size, and computation power), and sensitivity to location changes. In this paper, we will focus on the second characteristic of MobIS. We need to model both the variable locations a MobIS is expected to operate in, and the different behaviors the MobIS can adopt accordingly.

In the area of context modeling, the relation between context and its use is not clearly considered (e.g. [7], [8] and [9]). We believe in the tight complementary relation between variable behavior (both human and software ones) and context. When the relation between context and its use is omitted, we cannot answer questions like “*how do we decide the relevant context?*”, “*why do we need context?*” and “*how does context influence variable behavior?*”. Modeling context information is not a standalone activity, rather context has be elicited in conjunction with the analysis we do for discovering the alternative software behaviors. Salifu et al. [10] investigate the use of problem descriptions to represent and analyze variability in context-aware software; the work recognizes the link between software requirements and context information as a basic step to design a context aware system.

Software variability is a term commonly used to represent software provided with different behaviors, whose variants can be produced guaranteeing low costs, short time, and high quality [11]. Feature modeling is a well known modeling technique exploited by product line engineering to derive a tailored product from a family of possible products [12]. A mobile software is expected to select *autonomously* among the different alternatives it supports depending on the location settings. Lapouchnian et al. [13] propose techniques to design autonomic software based on an extended goal modeling framework, but the relation with the surrounding location is not focused on. A variant of this approach is proposed by the same authors in [14], where the emphasis is on variability modeling under the requirements engineering perspective, and the classification of intentional variability when decomposing a goal. In our work, we focus on variability of location, i.e. the unintentional variability, which influences the applicability and the efficiency of each goal satisfaction alternative.

Software agents are motivated by the need to act on behalf of users. An agent can perceive the environment where it lives, reason about it, and take decisions autonomously according to its beliefs. Agent-based software is composed of a set of social interacting agents that together form a multi-agent system (MAS)[15]. Agent-oriented software engineering (AOSE) methodologies is concerned with finding models and methods that lead to agent-based systems [16]. We believe that agent paradigm fits well for MobIS development, for both of them perceive the environment and autonomously act on behalf of humans.

Goal models, mainly adopted by KAOS [17] and Tropos [5, 6] agent-oriented methodologies, represent a paradigm shift from object orientation. While object-oriented analysis fits well to the late stages of requirement analysis, the goal-oriented analysis is more natural for the earlier stages [18], where we think it is also the natural place for modeling location. Goal models analyze, in a top-down way, a high level goal to discover the more specific sub-goals and tasks for satisfying that goal. Goal models allow for different alternatives to satisfy a goal, but do not specify in which locations each alternative can be adopted. We believe that alternative behaviors and location variability are complementary; supporting two alternative behaviors without specifying when to follow each of them rises the question *“why do we support two alternatives and not just one?”*. Conversely, considering location variability without supporting alternative behaviors rises the question *“what can we do if location changes?”*.

### 3 Motivating Example

To show how a MobIS can adapt its behaviour depending on the location, we describe an example concerning a client in a shopping mall. After entering the mall, the client inserts his/her smart-card into a card reader to get identified and authorized to take a PDA providing information about products. The company producing the PDA MobIS is required to develop it in a way it can adapt to different malls and support changes in a mall over time. One way of doing this is to consider all the variable mall settings, and design the software to autonomously choose an appropriate behaviour.

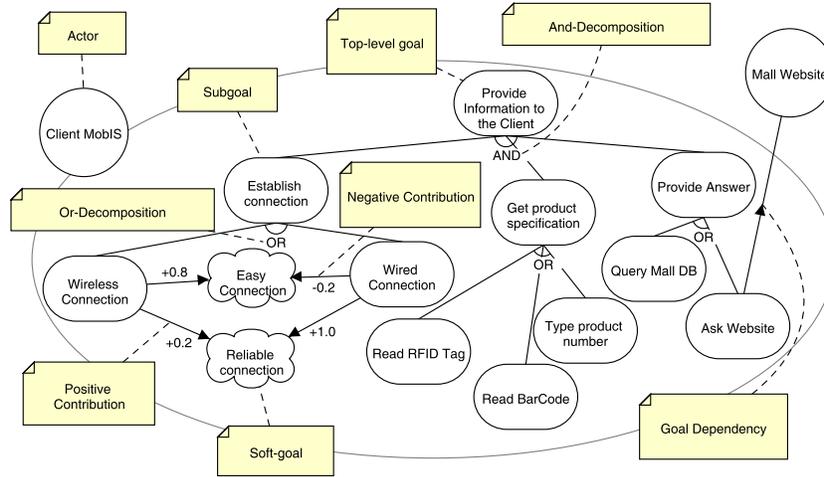
In the above introduced scenario, the PDA needs to get connected to the mall network, which can be either WiFi-based or a wired LAN whose access points are cable-based terminals. In the case of wired LAN the PDA has to show a demo that fits the client's language and expertise, and it should guide the client to the nearest free connection point. Positioning clients can be done in different ways, depending on the current location properties. An option is reading some positioning RFID tags located throughout the mall, using an RFID reader integrated to the PDA. Another way is using GPS positioning, if the client is in an open mall. If the mall contains some visible physical signs identifying the current zone, the client can be asked to provide what is written in the nearest visible sign. The scenario can also contain further variability aspects: when the connected client needs help about a product, the MobIS can either search for a suitable technician, query the mall database, redirect the PDA's browser to the mall website, or look for another client that can help.

The software company will consider different mall structures the MobIS supports. In each location the MobIS excludes some behaviours, and chooses one of the possible behaviours. The current location can also be analyzed to ensure that all MobIS goals are achievable. Furthermore, the existence of unachievable goals would motivate the location managers to change its properties (if possible), in order to overcome the problems that deny the satisfaction those goals.

## 4 Tropos for Location-based Variability

Goal-based modeling is intended to explain *why* a requirement is needed in the system to be. Goal analysis consists of an iterative refinement of a root *goal* through *And-/Or- decomposition* into *subgoals*, until identifying the software requirements needed to satisfy that root goal. Two main frameworks characterize the research in this area, namely KAOS [17] and Tropos [5,6]. In this paper we refer to the latter, since some concepts it provides fit well to modeling location-based variability: social relations (*dependencies*) between system *actors*, *contributions* to non-functional requirements (*soft-goals*), and its support to all of software development life cycle phases.

Fig.1 shows a part of one Tropos goal model to clarify this framework main concepts. Actors (*Client MobIS* and *Mall Website*) have a set of top-level goals (*Provide Information to the Client*), which are iteratively decomposed into sub-goals by and-decomposition (all subgoals should be achieved to fulfill the top goal) and or-decomposition (at least one subgoal should be achieved to fulfill the top goal). In Fig.1, the top-level goal is and-decomposed into *Establish Connection*, *Get product specification*, and *Provide Answer*; the goal *Provide answer* is or-decomposed into *Query Mall DB* and *Ask Website*. Soft-goals are goals for whose satisfaction there is no clear cut criteria (*Easy Connection* is a rather vague concept), and they are contributed either positively (0, +1] or negatively [-1, 0) by goals: *Wireless Connection* contributes positively (+0.8) to *Easy Connection*, while *Wired Connection* contributes negatively (-0.2) to *Easy Connection*. Goal dependencies represent situations where an actor cannot ful-



**Fig. 1.** A goal model labeled with the name of the basic concepts.

fill a goal by itself, but depends on another actor to fulfill it: actor *Client MobIS* depends on actor *Mall Website* for the achievement of goal *Ask Website*.

#### 4.1 Location-based Goal Model

In Tropos, the system is modeled as a set of social inter-dependent actors having goals, where actors can commit to strategy to satisfy their goals. Autonomous selection among alternative strategies requires representing the criteria an actor builds its decision upon. One alternative can be recommended in a certain location, while it can be even unapplicable in others. The selection criteria between alternatives is not explicitly modeled in the current Tropos goal model. We propose to integrate goal satisfaction intentional variability with location variability for deriving location-based software and enable automated reasoning.

When an actor tries to satisfy a goal, there can be several points where location specification is needed to take an appropriate decision between variable alternatives. We explain now our proposed list of goal model variability points:

1. *Location-based Or-decomposition:* Or-decomposition is the basic variability construct, but in current Tropos the choice of a specific Or-alternative is left to the actor intention, without explicitly considering location properties that can inhibit some alternatives. Example (from Fig.1): goal *Establish connection* can be achieved using *Wireless Connection* only if the mall has a wireless network and the client can access it.
2. *Location-based contribution to soft-goals:* the value of contributions to soft-goals can vary from one location to another. Example: the contribution from goal *Wireless Connection* to soft-goal *Reliable Connection* changes depending on the level of received signal: if the signal coming from the WiFi access

point is high, the contribution will be positive, while if the client is far from the WiFi access point and the signal level is poor, the contribution will be negative.

3. *Location-based dependency*: in certain locations an actor might not be able to satisfy a goal using any of its own strategies; in such case, the actor might delegate this goal to another actor that is able to satisfy it. Example: the MobIS can satisfy goal *Provide Answer* by fulfilling *Query Mall DB*; while if the database is offline and the mall website exists and has a mobile devices version, the MobIS can delegate the goal to another actor (*Mall Website*) browsing that website.
4. *Location-based goal activation*: an actor, when location settings change, might find necessary or possible triggering (or stopping) the desire of satisfying a goal. Self-activation is one of the main characteristics that distinguish an agent from an object [15], and Tropos needs to support it to better develop agent based systems. Example: if the MobIS has adopted the alternative *Wired Connection* for establishing connection, and while the client is getting to one cable-based terminal, the PDA detects a wireless signal, the goal *Wireless Connection* could be triggered to better satisfy the soft-goals.
5. *Location-based And-decomposition*: a sub-goal might (or might not) be needed in certain locations, that is some sub-goals are not always mandatory to fulfill the top-level goal in And-decomposition. Example: to satisfy the goal *Wired Connection*, the MobIS has first to show demo to client only if the client is using the system for the first time.

## 4.2 Location in Tropos

An information system designed to be self-reconfigurable needs to support different behaviors and to perform an autonomous selection among them according to some criteria, e.g. user preferences [19]. In our approach, the variability of MobIS is based on the different locations the system can adapt to, and the set of behaviors it can choose among. The autonomous location-based choice among alternatives and the applicability of each alternative requires a location model.

It is important to emphasize the subjective nature of defining a location. To clients, a mall employee is not more than a moving object if this employee does not satisfy any of the client objectives (like giving consultation about products). The mall network is not even noticeable if clients are not allowed to access it, or if the client has no need for it. Moreover, two different locations ( $L_1$  and  $L_2$ ) for an actor  $A_1$  can be considered as a unique location  $L_1 = L_2$  by an actor  $A_2$ . For instance, an actor with the objective of drinking coffee will see a mall with coffee machines different from a mall without them, while these two malls are equal for an actor who does not have such objective. Using Tropos concepts, we define location from the perspective of an actor as:

*The set of available actors and resources that can be employed to achieve actor goals.*



Fig.2 shows the goal model for a MobIS that is intended to operate in a shopping mall to help clients to get information about products. The top MobIS goal “*Provide Information to the Clients*” is top-down analyzed to get the MobIS requirements. We will mainly focus on the analysis of the sub-goal “*Establish Connection*”, and show how location properties can be defined on the variability points of the goal model. The location-based goal model is shown in Fig.2 and the location properties are expressed using labels on the variability points and will be formalized in the next section over the location model using Datalog $\neg$ .

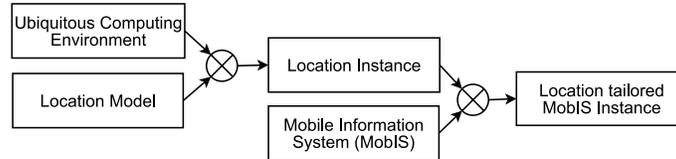
The goal “*Establish Connection*” can be achieved by fulfilling “*Wireless Connection*” when the mobile client is in a mall providing a wireless network (L1) or by fulfilling “*Wired Connection*” if the mall provides USB Cable-Based connection (L4). Wireless connection contributes positively to the soft-goal “*Easy Connection*”, while cable connection is less easy and gives a negative contribution to that soft-goal. Cable-based connection is almost always reliable; adopting this choice will satisfy to high degree the soft-goal “*Reliable Connection*”, while wireless connection is not always reliable. If the mobile user is close to the WiFi access point, the signal is high (L3) and the contribution to the soft-goal “*Reliable Connection*” is positive; the same contribution is negative if the signal is weak (L2).

When the MobIS decides to establish a wired connection, the system has to show a demo to the client explaining the connection process, and then guide client to the nearest connection terminal. The demo is needed only if the client uses the system for the first time (L5), so “*Show Demo*” is an optional goal. In order to show a demo, the information system should decide the demo language: this information can be provided “*Manually*” by the client, or “*Automatically*” if the client mother language or the mall default language is known (L6). The automatic language selection contributes positively to soft-goal “*User Comfort*”, while the manual selection contributes negatively. The system will select then between displaying an interactive demo, or a video like one. The “*Interactive demo*” goal will contribute positively to the soft-goal “*User Comfort*” if the client has good expertise using PDAs and the used PDA has a touch screen (L7), while it contributes negatively in the other case (L8).

Now we briefly describe the rest of the model. The system will guide client to the nearest cable connection terminal. The client current position has to be identified: it can be obtained either automatically through GPS, or reading an RFID positioning tag, or manually by asking user to type the content of the nearest positioning physical sign, or by showing the user an interactive (or picture) map to specify his/her current position. Then, the path to the nearest free terminal will be computed and the client be guided to that terminal. After getting connected, the client might ask for different information about products. The client will specify the product, and then the system will provide the required information. In the case the mall provides accessible database, the MobIS might query it, otherwise it might delegate this goal to the mall website actor (if the website has a mobile devices version).

## 6 Location Modeling

Satisfiability of location properties on the goal model will determine the appropriate MobIS goals satisfaction strategy. In our broad vision, the MobIS will be instantiated according to the current location. Technology advances make possible generating and communicating location description beyond user awareness. Ubiquitous computing and communication technology provide data that enables the derivation of a specific location instance from a location model. The location properties will be evaluated on this location instance to derive an adequate instance of a multi-behavioral MobIS. This vision is shown in Fig.3.



**Fig. 3.** The process of instantiating a location-tailored MobIS instance.

Location models are the basis for any kind of reasoning on location. We propose both a graphical location model, which can be drawn by designers, and an equivalent formal location model that enables automated checking of the location properties against an instance of the location model. Location-based goal analysis helps us to discover the entities that compose location and the relationships among them. In other words, location model is elicited by the location-based goal analysis.

### 6.1 Graphical Location Model

In this work, we rely on UML class diagrams to describe graphically locations. Class diagrams allow us to have a clear drawing and simple description of the relevant classes and relationships describing the location, and they are often used in conceptual modeling, and location rules can be expressed over them using OCL constraints [20].

Fig.4 shows the class diagram for the mall scenario proposed in Section 3; here we just describe its peculiarities, its interpretation should be easy by reading the figure. The specialization (is-a) relation is used to identify classes that share a common part but differ for some attributes or relations: for instance, a **Network** can either be **Wireless** or **Wired** depending on the type of access points it is composed of. Attributes can be used to define required information characterizing classes or relations: a WiFi access point (**WiFiAP**) is characterized by its position (**WPos**, representing the X and Y coordinates in the mall topology); the association **ConnectedAP** defines the clients connected to a particular **WiFiAP**, and every **Client** has an attribute **SigLevel** defining the signal strength of the

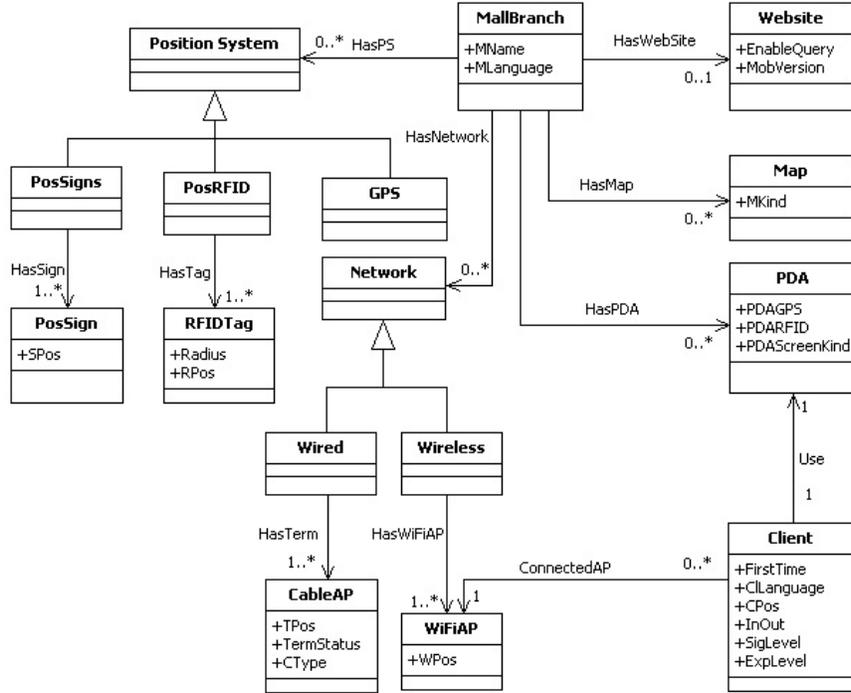


Fig. 4. Class diagram representing a location model.

WiFiAP the client is covered by it. Association is used to depict the relations between different classes: a `MallBranch` has zero to  $n$  `HasMap` relations to `Map`, a `Client` uses one `PDA` (which is in turn used by one `Client`).

## 6.2 Formal Location Model

The second modeling technique defines a formal location model, which is the basis to verify which location properties hold in a particular location instance (i.e. an instance of the formal location model). The formalization we propose is based on Disjunctive Datalog ( $\text{Datalog}^-$ ) [21], which satisfies our current need of formal knowledge representation. In particular, we use the syntax of the DLV solver [22]; it enables automated reasoning on knowledge, and its language adds some features like basic arithmetics support and cardinality check to the original  $\text{Datalog}^-$  language. We show part of the translation of the class diagram of Fig.4 to DLV in Table 1.

$\text{Datalog}^-$  is not based on classes, relations, and attributes, and we have hence defined translation rules to derive an equivalent representation based on predicates. We defined a  $\text{Datalog}^-$  predicate `Class`, which defines if a certain variable

MallBranch(A) :- Class(A), 1=#count{B:MLanguage(A,B), Bool(B)}, 1=#count{C:MName(A,C)}, 1=#count{W : HasWebSite(A,W), WebSite(W)}.
MallBranchPDA(A,B) :- MallBranch(A), HasPDA(A,B), PDA(B).
PDA(A) :- Class(A), 1=#count{B:PDAGPS(A,B)}, 1=#count{C:PDARFID(A,C), Bool(C)}, 1=#count{D:PDAScreenKind(A,D), ScreenKind(D)}.
Wired(A,B) :- Class(A), HasTerm(A,B), CableAP(B).
WiFiAP(A) :- Class(A), 1=#count{X,Y:WPos(A,X,Y), #int(X), #int(Y)}.
Wireless(A,B) :- Class(A), HasWiFiAP(A,B), WiFiAP(B).
Network(A) :- Wired(A,B).
Network(A) :- Wireless(A,B).
PDAConflict(A,F) :- Class(A), Class(B), PDA(F), Use(A,F), Use(B,F), A!=B
Client(A) :- Class(A), 1=#count{X,Y:CPos(A,X,Y), #int(X), #int(Y)}, 1=#count{C:InOut(A,C), Environment(C)}, 1=#count{L:CLanguage(A,L), Bool(L)}, 1=#count{D:FirstTime(A,D), Bool(D)}, 1=#count{F:Use(A,F), PDA(F)}, 1=#count{S:SigLevel(A,S), SignalStrength(S)}, 1=#count{P:ConnectedAP(A,P), WiFiAP(P)}, Use(A,G), not PDAConflict(A,G).

Table 1. Datalog $\neg$  formalization of part of the location class diagram.

is a class in the class diagram meaning. We exploited the #count aggregate operator, which is used both to check one-to-one or many-to-one relations and attributes; for example, a mall branch has exactly one attribute named `MName`, which represents the mall name. The representation of optional relations is based on the definition of auxiliary Datalog $\neg$  predicates: for instance, to express that a `MallBranch` has an optional relation `HasPDA`, we defined a binary predicate `MallBranchPDA` that represents a `MallBranch` connected to a `PDA` by `HasPDA` relation. Specialization is represented by defining more predicates (one for each subclass) for the same superclass: `Network` can be implied either by `Wireless` and by `Wired`.

We describe now in details the definition of `Client`, using class diagrams terms *relation* and *attribute* even though they are expressed as predicates in Datalog $\neg$ . A `Client` is a `Class`, which has exactly one attribute `CPos` identifying its current position; in turn, `CPos` defines the X and Y position in the mall topology using two integers. A `Client` also has exactly one `InOut` attribute identifying its position between “outdoor” and “indoor” values, a boolean `CLanguage` attribute which expresses if a default language has been set, a boolean `FirstTime` attribute that says if this is the first usage of the MobIS. Every `Client` is using one `PDA`, to which is connected by the relation `Use`. The `Client` also has an attribute `ExpLevel` that defines the expertise level in using a `PDA`. A certain `Client` cannot have any `PDAConflict` relation; it holds if there are two different users of the same `PDA`. A `Client` is associated to one `WiFiAP` through the relation `ConnectedAP`, and the signal strength is represented by the attribute `SigLevel`.

The location properties that were extracted informally by the location-based goal model of Fig.2, can be now formalized over the location model we already formalized. The formalization of the location properties (represented as labels in Fig.2) is shown in Table 2.

<p><b>(L1) A wireless network is available in the mall</b>  L1(A) :- Use(A,P), MallBranchPDA(M,P), NetworkedBranch(M,B), Wireless(B,C).</p>
<p><b>(L2-L3) There is a good (L2) or poor (L3) connection with a WiFi access point</b>  L2(A) :- Client(A), SigLevel(A,good).  L3(A) :- Client(A), SigLevel(A,poor).</p>
<p><b>(L4) An USB connection can be established</b>  L4(A) :- Use(A,P), MallBranchPDA(M,P), NetworkedBranch(M,B), Wired(B,C),  CType(C,usb), TermStatus(C,free).</p>
<p><b>(L5) The client uses the system for the first time</b>  L5(A) :- Client(A), FirstTime(A,true).</p>
<p><b>(L6) A default language has been chosen (by the mall or the client)</b>  L6(A) :- Client(A), CILanguage(A,true).  L6(A) :- Client(A), Use(A,P), MallBranchPDA(B,P), MallBranch(B), MLanguage(B,true).</p>
<p><b>(L7-L8) High (low, in L8) expertise in using the PDA and (not, in L8) touch screen</b>  L7(A) :- Client(A), Use(A,B), ExpLevel(A,low), PDAScreenKind(B,touch).  L8(A) :- Client(A), not L7(A).</p>
<p><b>(L9) There is a positioning system in the mall</b>  L9(A) :- Client(A), Use(A,P), MallBranchPDA(B,P), MallBranchPS(B,Q).</p>
<p><b>(L10) The client's PDA is in a location covered by the RFID tag signal</b>  L10(A) :- Client(A), Use(A,B), inCoverage(B,C), PDARFID(B,true),  MallBranchPDA(D,B), MallBranchPS(D,E), PosRFID(E,C).</p>
<p><b>(L11) The PDA has a connected/integrated GPS receiver</b>  L11(A) :- Client(A), Use(A,B), PDAGPS(B,true).</p>
<p><b>(L12-L13) The client is outdoor (L12) or indoor (L13)</b>  L12(A) :- Client(A), InOut(A,outdoor).  L13(A) :- Client(A), InOut(A,indoor).</p>
<p><b>(L14-L15) There are position signs (L14) or a map (L15) in the mall</b>  L14(A) :- Client(A), Use(A,B), MallBranchPDA(C,B), MallBranchPS(C,D), PosSigns(D,E).  L15(A) :- Client(A), Use(A,B), MallBranchPDA(C,B), MallBranchMap(C,M).</p>
<p><b>(L16-L17) Interactive map and PDA with touch screen (L16) or picture map</b>  L16(A) :- Client(A), Use(A,B), MallBranchPDA(C,B), MallBranchMap(C,M),  MKind(M,interactive), PDAScreenKind(B,touch).  L17(A) :- Client(A), Use(A,B), MallBranchPDA(C,B), MallBranchMap(C,M),  MKind(M,picture).</p>
<p><b>(L18) The mall branch has a mobile version of its website</b>  L18(A) :- Client(A), Use(A,B), MallBranchPDA(C,B), HasWebSite(C,W),  MobVersion(W,true).</p>

**Table 2.** The Location properties on the Goal model formalized in Datalog.

## 7 Analyzing Location-based Goal Models

This section presents three types of analysis for examining MobIS variability against a given location, and vice versa. A preliminary step consists of evaluating the validity of location properties at the variability points of the goal model (L1-L18 in Table 2) on an instance of the location model. This step can be done automatically using the DLV solver [22], and the result we get is those location properties that hold in the considered location instance.

We suppose the existence of two clients John and Mike, both located indoor and using a PDA with touch screen, GPS, and RFID reader. John has high expertise in using the PDA, while Mike’s expertise level is low. Their positions in the mall are different; it is the first time Mike uses the system. Mike has set a default language, John has not. We assume them to be in a mall named *SuperMall*, provided with position signs for directing people and a wireless network, while mall map and mall region language are not known.

In the following we propose three types of analysis that can be executed on location-based Tropos models: location-based goal satisfiability (LGS), location properties satisfiability (LPS), and preference analysis (PA).

### 7.1 Location-based Goal Satisfiability (LGS)

This kind of analysis is aimed to verify if a goal is achievable in the current location instance. In our example, with respect to client Mike, the goal “*Select Language*” cannot be achieved by choosing the sub-goal “*Automatically*”, because the evaluation of location property L6 is false.

The analysis can be performed using the goal reasoning algorithm proposed by Giorgini et al. [23] on the goal model restricted by the evaluation of the location properties. A strategy for evaluating satisfiability follows a top-down approach: starting from a top-level goal, we should check that all (at least one) sub-goals in and- (or-) decompositions can be achieved, or that the top-level goal can be achieved via a *makes* (+1.0) contribution from an achievable goal. In the mall MobIS, the top-level goal “*Provide Information to the Client*” can be fulfilled only if three subgoals are fulfilled; if the mall had neither a wireless connection nor a wired one, both L1 and L4 would be evaluated to false, and there wouldn’t be any way of fulfilling the goal “*Establish connection*”.

### 7.2 Location Properties Satisfiability (LPS)

This analysis checks if the current location structure is compliant with the MobIS goals. It is exploited to identify what is missing in a particular location where some top-level goals have been identified as unsatisfiable by LGS. When a goal cannot be satisfied, LPS will identify the denying conditions and suggest ways for solving the problem.

If wireless network is unavailable because there are not wireless access points working, the goal “*Establish connection*” will be unsatisfiable, and the problem is that both L1 and L4 evaluate to false ( $\neg L1 \wedge \neg L4$ ). Since there are not *makes*

contributions to higher level goals, the only way of allowing the satisfiability of the top goal is change the location in such a way that L1 or L4 ( $L1 \vee L4$ ) holds.

If we choose to enable L1, we have to examine the definition of L1 (Table 2); the first two predicates ( $Use(A,P, \_)$ ,  $MallBranchPDA(M,P)$ ) are true both for John and Mike, while the remaining two predicates are evaluated to false ( $NetworkedBranch(M,B)$ ,  $Wireless(B,C)$ ). Going into the details of the fourth predicate, we find that there are not wireless access points  $B$  such that  $HasWiFiAP(A,B)$  and hence the problem is identified. That is, we need to establish a  $HasWiFiAP$  relation with at least one  $WiFiAP$ .

### 7.3 Preferences Analysis (PA)

This type of analysis requires the specification of preferences over alternatives. Preferences can be specified using contributions to soft-goals as in [19]. We need this analysis in two cases:

1. When there are several alternatives to satisfy a goal: in our example, satisfying the goal “*Select Language*” for client Mike can be achieved both “*Manually*” and “*Automatically*” (L6 is evaluated to true because Mike has a default language). When mall administrator preferences give the soft-goal “*User Comfort*” high priority, the analysis will suggest the alternative “*Automatically*”.
2. When there is no applicable alternative: in this case, LPS might provide several proposals about the needed location modifications. The choice of a specific option can base on the preferences over soft-goals. If the mall has no connection available, and the mall administration preferences give “*Easy Connection*” a higher priority than “*Reliable Connection*”, the preference analysis will suggest to introduce a wireless connection (make L1 true) alternative to the mall structure.

## 8 Conclusions and Future Work

In this paper we showed the importance of modeling location settings as a basis to derive location-based MobIS behaviour. We also showed the subjective nature of defining location. From the perspective of an Tropos actor, a location is a set of available resources and other actors that are needed for achieving the goals of that actor. Information about current location is not necessarily needed by end users, but by the MobIS to decide which behaviour it should adopt. We used a variant of Tropos goal model that supports variability handling on the basis of location properties.

We exploited Datalog $\neg$  to formalize the location model and the location properties characterizing the variability points in the goal model. Formalization is needed because location properties are not generated to be simply shown, rather to enable the MobIS to reason about them. We presented three kinds of analysis that can be executed on the basis of location-based Tropos goal model.

Future work consists of expanding the proposed approach in various directions. We need to consider the meta-information the location information might have: for instance, some location entity attributes and relations are changeable over time, others are not. We also need to check how this can be presented formally and how the automated analysis will work with it. Moreover, we need to refine the analysis techniques by defining specific algorithms and test their efficiency.

## 9 Acknowledgement

This work has been partially funded by EU Commission, through the SERENITY project, by MIUR, through the MEnSA project (PRIN 2006), and by the Provincial Authority of Trentino, through the STAMPS project.

## References

1. Krogstie, J., Lyytinen, K., Opdahl, A., Pernici, B., Siau, K., Smolander, K.: Research areas and challenges for mobile information systems. *International Journal of Mobile Communications* **2**(3) (2004) 220–234
2. Pernici, B.: *Mobile information systems: infrastructure and design for adaptivity and flexibility*. Springer (2006)
3. Weiser, M.: The computer for the twenty-first century. *Scientific American* **265**(3) (1991) 94–104
4. Streitz, N., Nixon, P.: The disappearing computer. *Commun. ACM* **48**(3) (2005)
5. Yu, E.: *Modelling strategic relationships for process reengineering*. Ph.D. Thesis, University of Toronto (1995)
6. Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., Mylopoulos, J.: Tropos: An agent-oriented software development methodology. *Autonomous Agents and Multi-Agent Systems* **8**(3) (2004) 203–236
7. Yau, S., Liu, J.: Hierarchical situation modeling and reasoning for pervasive computing. *Proceedings of 3rd Workshop on Software Technologies for Future Embedded and Ubiquitous Systems (SEUS)* (2006) 5–10
8. Henriksen, K., Indulska, J.: A software engineering framework for context-aware pervasive computing. *PerCom* (2004) 77–86
9. Wang, X.H., Zhang, D.Q., Gu, T., Pung, H.K.: Ontology based context modeling and reasoning using owl. In: *PERCOMW '04: Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, Washington, DC, USA, IEEE Computer Society (2004) 18–22
10. Salifu, M., Nuseibeh, B., Rapanotti, L., Tun, T.: Using problem descriptions to represent variability for context-aware applications. *First International Workshop on Variability Modelling of Software-intensive Systems* (2007)
11. Pohl, K., Böckle, G., van der Linden, F.: *Software Product Line Engineering: Foundations, Principles, and Techniques*. Springer (2005)
12. Kang, K., Kim, S., Lee, J., Kim, K., Shin, E., Huh, M.: Form: A feature-; oriented reuse method with domain-; specific reference architectures. *Annals of Software Engineering* **5** (1998) 143–168

13. Lapouchnian, A., Yu, Y., Liaskos, S., Mylopoulos, J.: Requirements-driven design of autonomic application software. Proceedings of the 2006 conference of the Center for Advanced Studies on Collaborative research (2006)
14. Liaskos, S., Lapouchnian, A., Yu, Y., Yu, E., Mylopoulos, J.: On goal-based variability acquisition and analysis. Proc. 14th IEEE International Requirements Engineering Conference, Minneapolis, USA, Sep (2006) 11–15
15. Jennings, N.: On agent-based software engineering. *Artificial Intelligence* **117**(2) (2000) 277–296
16. Henderson-Sellers, B., Giorgini, P.: *Agent-oriented Methodologies*. Idea Group Pub. (2005)
17. Dardenne, A., van Lamsweerde, A., Fickas, S.: Goal-directed requirements acquisition. Selected Papers of the Sixth International Workshop on Software Specification and Design table of contents (1993) 3–50
18. Mylopoulos, J., Chung, L., Yu, E.: From object-oriented to goal-oriented requirements analysis. *Commun. ACM* **42**(1) (1999) 31–37
19. Liaskos, S., McIlraith, S., Mylopoulos, J.: Representing and reasoning with preference requirements using goals. Technical report, Dept. of Computer Science, University of Toronto (2006) <ftp://ftp.cs.toronto.edu/pub/reports/csrg/542>.
20. Warmer, J., Kleppe, A.: *The object constraint language: precise modeling with UML*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA (1998)
21. Eiter, T., Gottlob, G., Mannila, H.: Disjunctive datalog. *ACM Transactions on Database Systems (TODS)* **22**(3) (1997) 364–418
22. Leone, N., Pfeifer, G., Faber, W., Eiter, T., Gottlob, G., Perri, S., Scarcello, F.: The dlv system for knowledge representation and reasoning. *ACM Transactions on Computational Logic (TOCL)* **7**(3) (2006) 499–562
23. Giorgini, P., Mylopoulos, J., Nicchiarelli, E., Sebastiani, R.: Reasoning with goal models. Conceptual Modeling-ER 2002: 21st International Conference on Conceptual Modeling, Tampere, Finland, October 2002: Proceedings (2002)