UNIVERSITY
OF TRENTO

DIPARTIMENTO DI INGEGNERIA E SCIENZA DELL'INFORMAZIONE

38050 Povo – Trento (Italy), Via Sommarive 14
http://www.disi.unitn.it

LOCATION-BASED VARIABILITY FOR MOBILE INFORMATION SYSTEMS

Raian Ali, Fabiano Dalpiaz, and Paolo Giorgini

January 2008

Technical Report # DISI-08-008

Location-based Variability for Mobile Information Systems

Raian Ali, Fabiano Dalpiaz, and Paolo Giorgini

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

Abstract. Advances in size, power, and ubiquity of computing technology, sensors, and communication technology made possible the development of so-called mobile or nomadic information systems. A mobile information system has the potential to autonomously change its behaviour according to different location settings. Variability of location and system behaviour is a central feature of such new generation of information systems. This paper stresses the importance of modeling and analyzing variability of location as a basis for variability of software. We describe graphical and formal techniques to model location information, show their usage in conjunction with the goal-oriented framework *i**/Tropos, and propose three analysis techniques on location-based goal models.

1 Introduction

Advances in computing and communication technology have led to the growth of interest in Mobile Information Systems (hereafter MobIS); MobISs emphasize mobility concerns often not considered by desktop systems [1, 2]. Computing tends to be an integrated part of the environment instead of being an external entity directed by humans; interaction with computers is getting hence as minimal as possible [3]. Besides computing ubiquity, the 21st century computing [4] should have a core "mental" part: computing systems act on behalf of humans executing tasks without prompting them and receiving their explicit requests. Variants or parts of this vision are Ambient Intelligence, Disappearing Computer, Pervasive Computing, Ubiquitous Computing, Calm Technology, Wearable Computing, and so on.

Developing environment-aware computing systems is now feasible, but some obstacles have still to be overcome. The proliferation of pocket devices (PDA, 3G mobile phones), the availability of advanced sensors (RFID, fingerprint readers), and the modern communications techniques (WiFi, Bluetooth) relieve humans of many iterative tasks that are typical in desktop scenarios (e.g., password typing, cable-based connection establishment) and open the door to more innovative software usage scenarios. New challenges for information systems engineering arise, since software is given more responsibility, and it can now actively support the decision making process. However, as any new technology, methods and models need to be developed or at least adapted to cope with the new achievable innovative requirements [1, 5].

Nomadic user needs smarter information systems, able to adapt their behaviour without human intervention. Systems should be able to reason about the surrouding location, which includes the user itself, and adapt autonomously their behaviour to location changes. What we need then is to model and analyze the variable locations users can be part of, and define how location influences the behaviour of the MobIS.

Software agents [6] 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). Agent-oriented software engineering (AOSE) [7] is concerned with finding models and methods that lead to agent-based systems. We believe that a tight relation between MobIS requirements and the agents paradigm exists, for both of them perceive the environment and autonomously act on behalf of humans.

In this paper, we use an agent-oriented framework, namely *i**/Tropos [8, 9], to model the different behaviours a MobIS can adopt. We define the points of variability in *i**/Tropos goal model, together with the location properties needed to support the decision making process. The MobIS will use the location information and interpret them to switch to a proper behaviour. The location data conceptual model is designed graphically (using EER modeling) by designers, and then formalized using Datalog¬[10] to allow automated reasoning about MobIS behaviour and location design.

The paper is structured as follows: in Section 2 we review related works. In Section 3 we describe the relation between location and MobIS behaviour adaption. In Section 4 we discuss the usage of *i**/Tropos goal-oriented framework for representing the variable behaviours a MobIS can switch to depending on location properties; we use *i**/Tropos conceptualization to give a subjective location definition. In Section 5 we describe graphical and formal modelling of location. In Section 6 we show location-based goal modeling. In Section 7 we describe three kinds of analysis, and in Section 8 we draw conclusions and present future work.

2 Related Work

Software variability is a term commonly used to represent software provided with different behaviours, whose variants can be produced in accordance to the stakeholder's requirements and guaranteeing low costs, short time, and high quality [11]. Feature models are a well known modeling technique that is exploited by product line engineering to derive a tailored product from a family of possible products [12]. The main concern of product family engineering and feature models is allowing software variability, while the choice of the actual product to develop is made by designers. A MobIS has to select autonomously between 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 clearly shown and analyzed. 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, with a focus on the classification of variability concerns. One of those concerns refers explicitly to location, but no actual solution is currently available, to our knowledge.

In the area of context-aware systems, context is treated apart from software requirements. We believe that requirements should be the pivot for deciding and filtering what are the relevant context information. In [15], the authors propose the usage of ontologies to model situation, deriving first-order logic rules to represent it. Wang et al. [16] propose CONON, an upper ontology for representing general context, while domain-specific ontologies can be derived specializing CONON. Both these approaches lack of links to software requirements. Salifu et al. [17] investigate the use of problem descriptions to represent and analyze variability in context-aware software; their work recognizes the link between software requirements and context information, but they do not really suggest a solution to the problem. We believe that getting context information is not a goal by itself, and this information should not be modeled using a separate model, rather it is derived information (*views*) over a model of what exist.

Goal models, mainly adopted by KAOS [18] and *i**/Tropos [8, 9], represent a paradigm shift from object orientation. Goal models allow for different alternatives to satisfy a goal, and this is of great importance for MobISs, since the system needs to adapt its behaviour with respect to the current location. Without alternatives, considering location variability would be senseless. The goal-oriented *i**/Tropos framework is aimed at developing agent-based systems, and hence it fits well to MobISs that perceive the location around and act accordingly.

3 Location-based variability: a 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 such 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 DB, redirect the PDA's browser to the mall website, or search 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's managers to change its properties (if possible), in order to overcome the problems that deny those goals.

4 Location-based *i**/Tropos

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 [18] and *i**/Tropos [8, 9]. In this paper we refer to the latter, since some concepts it provides fit well to modeling location-based variability: *Or-decomposition*, social relations (*dependencies*) between system *actors*, and *contributions* to non-functional requirements (*soft-goals*).

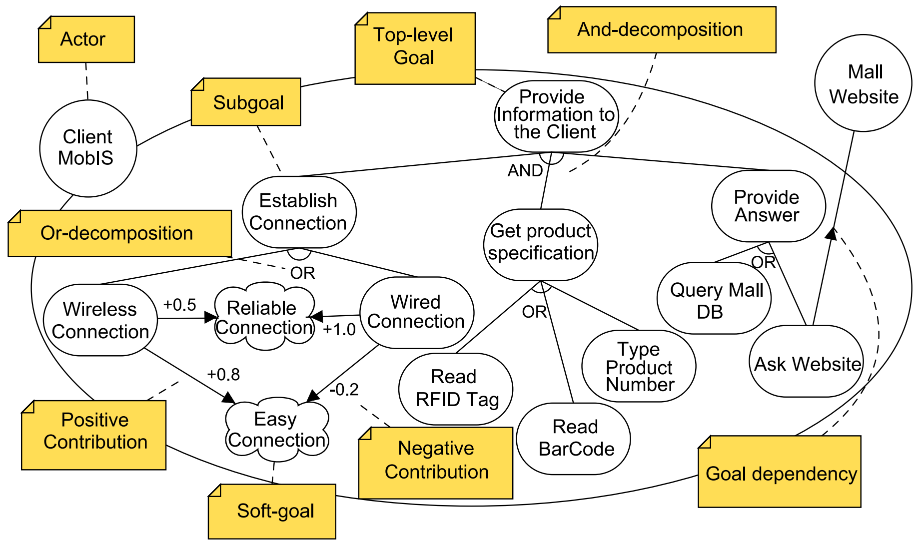

Fig. 1. A goal model labeled with the name of the basic concepts.

Figure 1 shows a part of an *i**/Tropos goal model to clarify this framework's 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 subgoals 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 Figure 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 fulfill 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 Handling location-based variability using *i**/Tropos

In *i**/Tropos, the system is modeled as a set of social inter-dependent actors having goals, where actors can commit to plans to satisfy their current goals. Autonomous selection among alternative plans 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 *i**/Tropos goal model. We propose to enrich *i**/Tropos goal model with needed location properties at its variability points, to enable various automated analysis.

Location-based *i**/Tropos framework is an extension of *i**/Tropos that adds location properties at the following variability points:

1. *Or-decomposition*: Or-decomposition is the basic variability construct, but in current *i**/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 Figure 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. *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 (+0.8, for instance), while if the client is far from the WiFi access point and the signal level is poor, the contribution will be negative (-0.5).
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. *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 [19], and *i**/Tropos needs to support it since it is an agent-oriented framework. 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. *And-decomposition*: a sub-goal might (or might not) be needed in certain location, 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 a descriptive demo to client only if the client is using the system for the first time.

After enriching the goal model with the needed location properties, a variety of analysis become possible:

1. Evaluating if a goal is achievable in the current location.

2. Selecting the best way to achieve a goal when there are more than one applicable alternatives. The decision making can be done:
   (a) on the basis of contribution (possibly location-based) to some soft-goals;
   (b) based on preferences, that in turn can be expressed on the basis of soft-goal (as proposed by Liaskos in [20]): soft-goals act as an engine for the variability rationale;
3. Knowing which changes are needed in the location to satisfy all objectives. In this way we reveal the requirements and the design of the location that is intended to satisfy a variety of MobIS goals.

Beyond the role of defining software requirements and giving a rationale for them, location-based goal analysis plays other roles. Defining the needed location properties at each variability point will guide the software production process with respect to the actual location instance. Moreover, this analysis will elicit what location information are needed. The location information can be obtained from different sources (sensors, RFID tags, DBs, and so on) and then they can be communicated to the MobIS using different communication technologies (WiFi, Bluetooth,..). Consequently, knowing what location information that have to be obtained and provided will reveal what infrastructure is needed. In other words, the location information needed by MobISs will be the basis for engineering a location infrastructure enabling the retrieval and distribution of information, so we can talk about location engineering as well.

4.2 Location in *i**/Tropos

An information system designed to be self-reconfigurable needs to support different behaviors and should perform an autonomous selection among them according to some criteria, e.g. user preferences [20]. The variability of a MobIS consists of the scenarios the system can adapt to, and the set of behaviors (supported by features) it can choose among. The autonomous choice among the alternatives and the applicability of each alternative requires a model of the relevant location assets.

It is important to emphasize the subjective nature of defining a location. To clients, a mall employee is no 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 *i**/Tropos concepts, we define location from the perspective of an actor as:

The set of available actors and resources that can be used to achieve an actor's goals.

From the perspective of an actor, the availability of a resource means that the resource exists and at the same time it can be exploited to achieve some objectives. Resources can be physical resources, information, or even an actor skills. A resource in *i**/Tropos modeling language is an unintentional entity, that has no meaning if it is not relevant to some goals. Available actors are similarly those actors an actor can depend on for goals to be achieved or resources to be furnished or tasks to be executed.

5 Location modeling

The location information will be the input that guides the software derivation process: the MobIS will be instantiated according to the current location instance. Technology advances make possible generating and communicating this information beyond user awareness. Pervasive computing and communication technology provide data that enable the derivation of a specific location instance from a location model, and this location instance will contribute deriving an adequate instance of a multi-behavioral MobIS as shown in Figure 2.

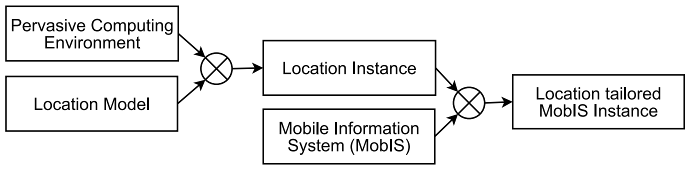

Fig. 2. The process of instantiating a location-tailored MobIS instance.

MobISs have to define a set of location properties at the points where a location-based behavior is needed. Location properties are expressed as *views* over a location model, and they can be either true or false depending on the location instance. The location data conceptual model is the basis of any kind of reasoning on location. We propose both a graphical model, which is used in software design phases, and an equivalent formal model that enables automated checking of the location properties against the current location instance.

5.1 Graphical Location model

We used Entity-Relationship model to describe locations. ER diagrams allow us to have a clear drawing and simple description of the relevant entities and relationships that form the location. Particularly, we use the Extended Entity-Relationship (EER) model, which extends ER adding some concepts like the is-a relation.

Figure 3 shows the EER 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 entities 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 entities or relations: a WiFi access point (`WiFiAP`) is characterized by its position (`WPos`, representing the X and Y coordinates in the mall topology); the relation `ConnClient` defines the clients connected to a particular `WiFiAP`, and it has an attribute `Signal` identifying the signal level. Relations connect entities and can have cardinality restrictions: a `MallBranch` has zero to *n* `HasMap` relations to `Map` entities, a `Client` uses exactly one `PDA` (which is in turn used by exactly one `Client`). It is worth noting that we represent only entities and relations that are meaningful for the depicted location: a `PDA` could also be unused

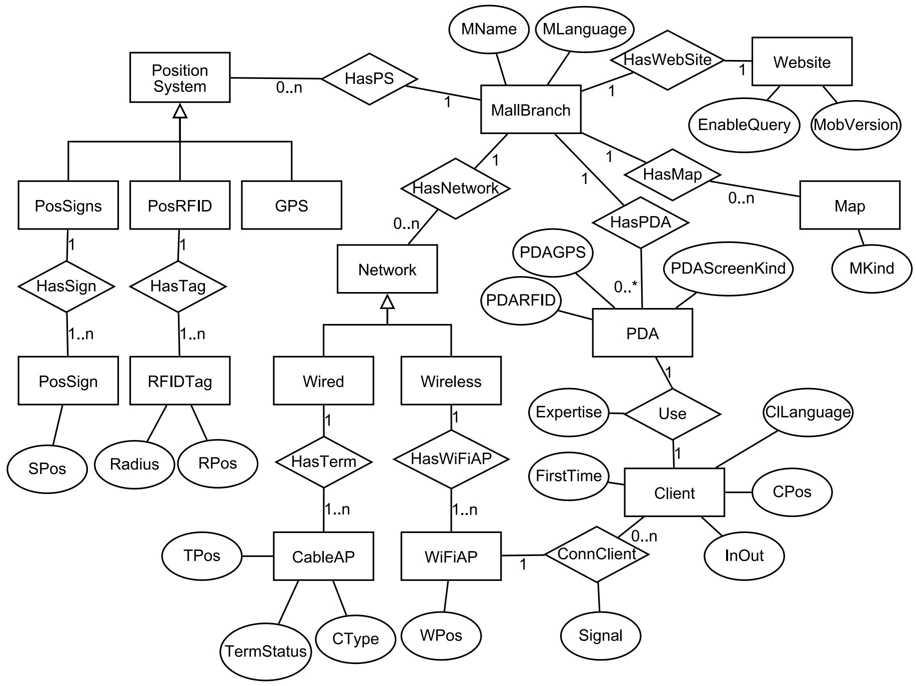

Fig. 3. EER diagram representing a location data conceptual model.

(i.e. without an `Use` relation with a `Client`), but in that case it would be irrelevant for the developed MobIS.

5.2 Formal location model

The second modeling technique defines a formal location model, which is the basis to verify if 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 (Datalog¬) [10], which is suitable for formal knowledge representation. In particular, we use the syntax of a Datalog¬-based solver named DLV [21]; it enables automated reasoning on knowledge, and its language adds some features like basic arithmetics support and cardinality check to the original Datalog¬ language. We propose part of the translation of the EER diagram of Figure 3 to DLV in Table 1.

Datalog¬ is not based on entities, relations, and attributes, and we have hence defined translation rules to derive an equivalent representation. We defined a Datalog¬ predicate Entity, which defines if a certain variable is an entity in the EER 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¬ predicates: for instance, to express that a `MallBranch` has an optional relation `HasPDA`, we defined

MallBranch(A) :- Entity(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) :- Entity(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) :- Entity(A), HasTerm(A,B), CableAP(B).
WiFiAP(A) :- Entity(A), 1=♯count{X,Y:WPos(A,X,Y), ♯int(X), ♯int(Y)}.
Wireless(A,B) :- Entity(A), HasWiFiAP(A,B), WiFiAP(B).
Network(A) :- Wired(A,B). Network(A) :- Wireless(A,B).
PDAConflict(A,F) :- Entity(A), Entity(B), PDA(F), Use(A,F,E), Expertise(E), Use(B,F,G), Expertise(G), A!=B.
Client(A) :- Entity(A), 1=♯count{X,Y:CPos(A,X,Y), ♯int(X), ♯int(Y)}, 1=♯count{C:InOut(A,C), Environment(C)}, 1=♯count{L:ClLanguage(A,L), Bool(L)}, 1=♯count{D:FirstTime(A,D), Bool(D)}, 1=♯count{F:Use(A,F,E), PDA(F), Expertise(E)}, Use(A,G,E2), not PDAConflict(A,G).
ConnClient(A) :- Client(A), 1=♯count{B,C:Signal(A,B,C), WiFiAP(B), Strength(C)}.

Table 1. Datalog¬ formalization of part of the EER conceptual model.

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 subentity) for the same superentity: `Network` can be implied either by `Wireless` and by `Wired`.

We describe now in details the definition of `Client`, using EER terms *relation* and *attribute* even though they are expressed as predicates in Datalog¬. A `Client` is an `Entity`, 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 `ClLanguage` 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` has exactly one `PDA`, which is connected by the relation `Use`, which in turn has an attribute `Expertise` representing the `PDA` expertise level of the `Client`. Lastly, a `Client` cannot have any `PDAConflict` relation; it holds if there are two different users of the same `PDA`.

6 Location-based *i**/Tropos Goal Model

Analyzing a goal will lead finally to discovering what should be done for satisfying that goal. Moreover, goal analysis will reveal what location information needed to decide among alternatives at the variability points of the goal model identified in Section 4.1

A MobIS intended to operate in a shopping mall is concerned with helping clients to get information about products. As shown in Figure 4, the top MobIS goal *"Provide Information to the Clients"* is top-down analyzed to get the mobile system requirements. We will mainly focus on the analysis of the sub-goal *"Establish Connection"*, in order to define the location properties to be evaluated in the variability points of the goal model, in addition to the MobIS requirements themselves. The location-based

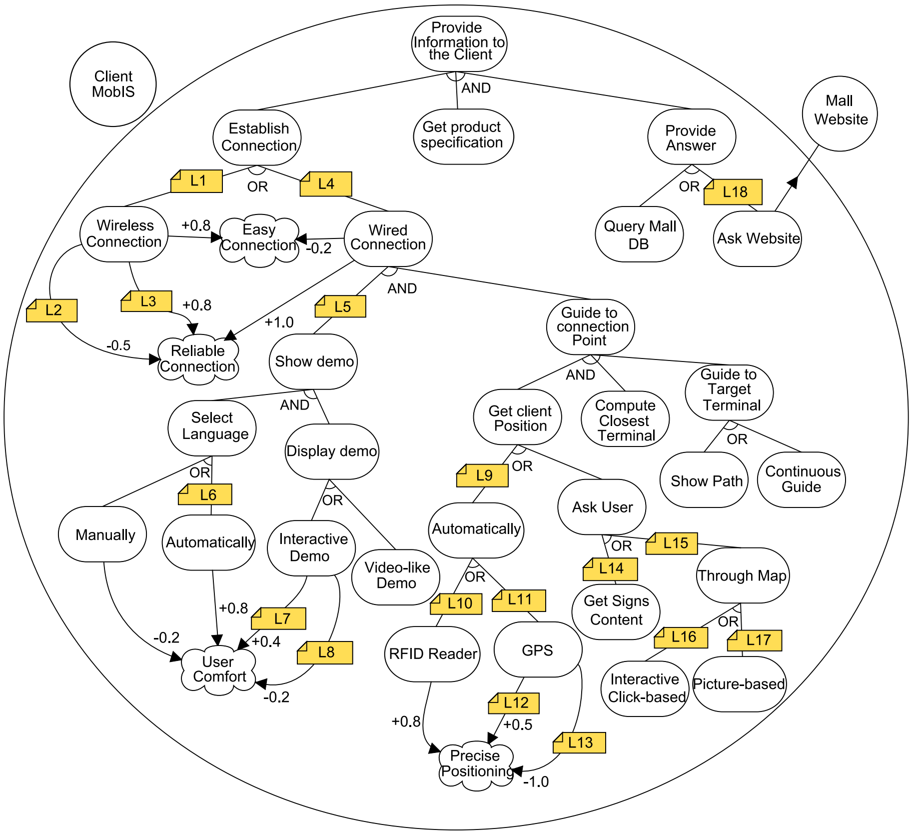

Fig. 4. A Location-based Goal Model.

goal model is shown in Figure 4 and the location properties needed are expressed using labels on the variability points and formalized using Datalog¬ in Table 2.

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

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

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

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 otherwise (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 DB, the MobIS might query it, otherwise it might delegate this goal to the mall website actor (if the website has a mobile devices version).

7 Analyzing location-based *i**/Tropos models

This section presents three types of analysis for examining MobIS variability against the current location instance, 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 the current location instance. This step can be done automatically using the DLV solver, 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 *i**/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. [22] 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 [20]. 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 *i**/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 *i**/Tropos goal model that supports variability handling on the basis of location properties.

We exploited Datalog¬ 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 *i**/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.

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. Streitz, N., Nixon, P.: The disappearing computer. Commun. ACM **48**(3) (2005)
4. Weiser, M.: The computer for the twenty-first century. Scientific American **265**(3) (1991) 94–104
5. Krogstie, J.: Requirement engineering for mobile information systems. Proc. of International Workshop on Requirements Engineering: Foundation for Software Quality (Interlaken, Switzerland) (2001)
6. Wooldridge, M.: Agent-based software engineering. Software Engineering. IEE Proceedings **144**(1) (1997) 26–37
7. Henderson-Sellers, B., Giorgini, P.: Agent-oriented Methodologies. Idea Group Pub. (2005)
8. Yu, E.: Modelling strategic relationships for process reengineering. Ph.D. Thesis, University of Toronto (1995)
9. 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
10. Eiter, T., Gottlob, G., Mannila, H.: Disjunctive datalog. ACM Transactions on Database Systems (TODS) **22**(3) (1997) 364–418
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. 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
16. 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
17. 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)
18. 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
19. Jennings, N.: On agent-based software engineering. Artificial Intelligence **117**(2) (2000) 277–296
20. 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.
21. 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
22. 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)