2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing

A Formal Model for Advanced Physical Annotations

Ahmad A. Alzahrani, Seng W. Loke and Hongen Lu
Department of Computer Science and Computer Engineering
La Trobe University, Victoria 3086, Australia
{aaalzahrani@students.latrobe.edu.au, s.loke@latrobe.edu.au, h.lu@latrobe.edu.au}

***Abstract*— Mixing the virtual world and the physical world seamlessly has become a phenomenal mobile service to users, providing new directions in pervasive computing. There are many variants for this mix such as Physical Annotations (PAs) and Mixed Reality (MR). There are many uses of PAs and MR such as education, entertainment, shopping, tourism and more. Many platforms have been developed to provide this service. However, there is still no standard or formal definition for PAs which aims to consolidate and extend the possibilities with PAs. The aim of this paper is to explore the concept of the Physical Annotation and to provide a formal model for it. The paper analyzes and applies the formal model to some of the existing major systems that are used for PAs. Then, we propose a generic PA system architecture and illustrate our model using a scenario concerning a shopping center.**

Keywords: Physical Annotation; formal model; augmented reality.

I. INTRODUCTION

A Physical Annotation (PA) is a type of digital information which is associated with one or more physical entities, which may be a place, a small object, or even a person. A PA aims to enable the interaction between users and inanimate objects or geographical places, which will make them more intelligible and responsive to the users. A PA could be private/personal or shared with other users. The aim of the PA is to add more explanations and descriptions about the entity or entities. PAs can be used for different purposes such as commercial advertising, tourist guiding or even as personal memories. A PA has many features, and can be structured into three parts, the first one concerns the annotation itself. The second part is about the target entity. The third part is the link between the annotation and the target. Some researchers such as [1-4] have discussed PA challenges such as PA structure, and ways to access, present and edit PAs. Therefore, this type of annotation has grown rapidly in the last few years. However, as demand for more complex PA structures and applications grow, a formal model can help provide clarity and a basis for future developments

The aim of this paper is to provide a formal model for physical annotations, in the spirit of [12]. We show that such a formal model is not only for analyzing PA systems and describing PA systems, but also provides a basis for the generic architecture of PA systems.

The rest of this paper is organized as follows. Section 2 presents a formal model for PAs. Section 3 analyzes existing PA systems using the attributes in our formal model. Section 4 discusses a generic architecture for PA systems based on the formal model and discusses scenarios for a PA system based on the architecture and our formal model. Section 5 reviews related work, and Section 6 concludes with future work.

II. A FORMAL MODEL OF PAS

We model a PA as comprising three main parts: 1) the annotation part, which includes the content, format, type, etc; 2) the physical entity (the target) being annotated such as a location, or a small object; and 3) the link between the annotation and the annotated target, which includes the conditions of the relationship, anchor properties, and so on.

1. The annotation

The annotation part comprises the following properties: the annotation ID, annotation type, content type, the annotation content itself, ussers, groups, author category, and user cooperation, as explained below.

1.1 Annotation ID

This ID is a unique identifier for each annotation. The ID should also imply whether the annotation is standalone or dependent which will be discussed in the mapping property.

Definition 1.1 (annotation identifier)

We define $\boldsymbol{AID}$ as a set of annotation identifiers, and $\mathbf{aID} \in \boldsymbol{AID}$ denotes an annotation identifier.

1.2 Annotation Type

This property concerns the purpose of the annotation. There are many purposes for creating annotations; therefore, based on those purposes, there is a need to classify these annotations. Examples of annotation purposes/types are educational, commercial, health, security and more. These types will contribute to providing a better system for the end user by allowing users to choose annotations based on purpose/type, thereby reducing overload on the mobile (other criteria for filtering annotations can be used as discussed later).

Definition 1.2 (annotation type)

We define $\boldsymbol{AT}$ as a set of annotation types, and $\boldsymbol{at} \in \boldsymbol{AT}$ denotes an annotation type. The set $\boldsymbol{AT}$ = {education, tourism, health, security, entertainment, commercial}. (Other types can be added but this is an initial proposal based on observing PA application areas.)

1.3 Content Type (Media Type)

This property describes the media type of the annotation content, whether it is text, sound, picture, video or a combination of one or more media types. The way of representing annotations will determine the mechanism for accessing the annotations, and a suitable device should be used to access it. For example, if the annotation is a video

978-0-7695-4612-4/11 $26.00 © 2011 IEEE
DOI 10.1109/DASC.2011.50

clip, the user should use a mobile that can play videos; also, the mobile connectivity should be sufficient for downloading such videos.

Definition 1.3 (content type (media type))

We define $\boldsymbol{MD}$ as a set of content media types, and $\boldsymbol{md} \in \boldsymbol{MD}$ denotes a media type that is allowed for an annotation. The basic $\boldsymbol{MD}$ set captures common media types:

$\{text, picture, sound, video, mixed$ media$\}$.

1.4 Annotation Content

This property contains the actual information and the real content of the annotation. Based on all previous properties, the content here should only be the information created by the user and it should be short and meaningful.

Definition 1.4 (annotation content)

$\boldsymbol{AC}$ is a set of possible annotation contents (e.g., as defined by an XML schema, or a set of video clips) and $\boldsymbol{ac} \in \boldsymbol{AC}$ denotes a piece of annotation content (e.g., a document conforming to the XML schema, or a video clip belonging to the pre-specified set). .

1.5 Users

This property includes all the system users. The user could be an author who is able to annotate an entity, or could be a reader who is allowed to access the annotation.

Definition 1.5 (users)

Let $\boldsymbol{USER}$ be a set of users, and $\boldsymbol{user} \in \boldsymbol{USER}$ denotes a generic user.

1.6 Group

The aim of this property is to enable organization of users and classify them into groups based on their interests. Moreover, the author will have the choice to address his/her annotation to a particular group. For example, there could be educational annotations which were addressed to high school students, so the only users who can access these annotations are such students. Otherwise, the user will be denied access to them, or even will not be able to discover/see them. A user may also belong to more than one group.

Definition 1.6 (groups)

$\boldsymbol{GR} \subseteq 2^{\boldsymbol{USER}}$ is a set of groups for all users. And $\boldsymbol{gr} \in \boldsymbol{GR}$ denotes one group of users.

1.7 Author category

This property describes the identity of the author who can create, delete or modify the annotation. The author could be an authorized person to annotate an object (having the category of "official"). For example, a shopping centre may authorize only some staff to annotate the shops, the advertising and so on. The author may be a normal user, so s/he may give his/her feedback or recommendation to other users about that object. The author may also be a friend, so s/he makes annotations for his/her friends. The author can be the user her/him-self, so s/he may like to retrieve the annotations that were made by him/her only. This property gives also the reader the chance to choose the source of the annotations based on his/her preferences. The non mutually exclusive author categories are {official, normal users, friend, self}.

Definition 1.7 (authors)

We define $\boldsymbol{AU}$ as a set of authors of annotations, and $\boldsymbol{au} \in \boldsymbol{AU}$ denotes a generic author, and $\boldsymbol{AU} \subseteq \boldsymbol{USER}$.

1.8 User cooperation

This property distinguishes a user as the author of the annotation with authority over read/write access of the annotation. The author of an annotation has the right to authorize other users to allow them to access and read the annotation. This privilege can be private, shared or public. The annotation could be private for the user him/her-self only, shared with his/her friends (or particular groups of users) or could be available for all users. This propriety is very important in order to address a specific purpose of an annotation for a specific group.

Definition 1.8 (user cooperation)

Let $\boldsymbol{COP} = \{private, shared, public\}$ be a set of user cooperation types, and $\boldsymbol{cop} \in \boldsymbol{COP}$ denotes a cooperation type. Let us define the following relation: strict ordering relation denoted by $\prec$ is {(private, shared), (private, public), (shared, public)}.

Based on the above definitions, we can now define an annotation as follows.

Definition 1 (annotation)

An annotation $\boldsymbol{ann}$ is an 8-tuple of the following form:

$$\boldsymbol{ann} = < \boldsymbol{aID}, \mathbf{at}, \mathbf{md}, \mathbf{ac}, \mathbf{au}, \mathbf{user}, \mathbf{gr}, \mathbf{cop} >$$

where:

$\boldsymbol{aID}$ is an annotation identifier, i.e. $\boldsymbol{aID} \in \boldsymbol{AID}$, **AID** is the set of annotation identifiers,

$\boldsymbol{at}$ is the annotation type, i.e. $\boldsymbol{at} \in \boldsymbol{AT}$,

$\boldsymbol{md}$ is the media type, i.e. $\boldsymbol{md} \in \boldsymbol{MD}$,

$\boldsymbol{ac}$ is the content type, i.e. **ac** $\in \boldsymbol{AC}$,

$\boldsymbol{au}$ is the annotation's author, i.e. $\boldsymbol{au} \in \boldsymbol{AU}$,

$\boldsymbol{user}$ is the annotation's user, i.e. $\boldsymbol{user} \in \boldsymbol{USER}$,

$\boldsymbol{gr}$ is the user's group, i.e. $\boldsymbol{gr} \in \boldsymbol{GR}$, and

$\boldsymbol{cop}$ is the user cooperation type, i.e. $\boldsymbol{cop} \in \boldsymbol{COP}$.

2. The target

2.1 Target ID

This property describes the unique identifier of the annotated entity. It aims to give each entity a unique identifier in order to distinguish annotated entities from each other. The target identifier should be informative in the sense of encoding the structure of the entity in the physical world. For example, if the target is a room in a building, the room identifier should contain the building identifier and the floor identifier. This technique will help the annotation system to understand the location of the entity in a hierarchical structure (which can be used for filtering annotations according to the user's vicinity). For example, the user is given the option to retrieve all the annotations of this room, and also to retrieve all the annotations of the floor, or all the building. The target identification should also contain some description of the nature of the entity. For example, if the building is a shopping centre, the identifier should contain some symbol that implies the type of the entity. This aims to determine the possible annotations that could be associated with this entity, and also to prevent attaching unsuitable annotations to it.

Definition 2.1 (target identifier)

We define $\boldsymbol{t\_id}$ as a set of target identifiers, and $\mathbf{t\_id} \in \boldsymbol{T\_ID}$ denotes a target identifier.

2.2 Target Type

This property aims to describe the nature of the annotated entity. The entity can be a location, a small object such as a cup, or even a person. Defining and classifying the target's type will provide better understanding of the possible annotations that could be associated with that entity. Each entity has its own features; however, in general, all of them will have some similarities in characteristics and features. For example, almost all buildings in a CBD will have the same properties such as that they all have floors, addresses, and geographic locations. This will help to create abstract annotations which may be used for different entities. We can build a binary tree to create an abstract type to describe the physical entity. We start by creating three abstract types: location, small object, and person. Then, each one of these types will be divided into more specific groups. For example, the location type can be divided into commercial, residential and so on. The target type classification will contribute towards reducing the huge numbers of annotations, so the user can chose his/her preference target type and avoid other types. Below is a generic set of target types.

Definition 2.2 (target type)

We define $\boldsymbol{T\_T}$ as a set of target types, and $\boldsymbol{t\_t} \in \boldsymbol{T\_T}$ is a target type. The set $\boldsymbol{T\_T}$ = {location, object, person}.

Definition 2 (target)

A target is a pair comprising type id ($\boldsymbol{T\_ID}$) and the target type ($\boldsymbol{T\_T}$), so the target is a pair as follows.:

$$\boldsymbol{t = < t\_id, t\_t >}$$

where:

$\boldsymbol{T}$ is a set of all targets, and $\boldsymbol{t} \in \boldsymbol{T}$ is a target,

$\boldsymbol{T\_ID}$ is a set of all possible target IDs, and $\boldsymbol{t\_id} \in \boldsymbol{T\_ID}$ is a target identifier, and $\boldsymbol{T\_T}$ is a set of the annotated entity types or target types (e.g., location or a small object), and $\boldsymbol{t\_t} \in \boldsymbol{T\_T}$ denotes a target type.

3. The link

A. Association

This property describes how an annotation is related to the target entity(-ies), whether it describes a feature of the entity, or provides secondary information which is related to the entity. A feature of the entity is about annotating the entity itself such as a building name, the history of that building, or any description about the building itself. Therefore, the association of this annotation is called *strong*. An annotation may also have a partial or *weak* relation to the annotated entity's features. For example, when a user describes something happening near that building, or information about someone s/he has met at that building. So the annotation is not about the entity itself, but about something involving it.

Definition 3.1 (association)

We define $\boldsymbol{ASSO}$ to be a set of association types, and $\boldsymbol{asso} \in \boldsymbol{ASSO}$ corresponds to one of the association elements which are {strong, weak}.

B. Accessibility Type

This property describes how the annotation is accessed and retrieved. The way of accessing the annotation could be direct or indirect. In direct access of an annotation, the user can read the annotation directly from the object itself without needing Internet connectivity. The NFC technology is an example of retrieving the annotation directly, where the annotation can be stored in an RFID tag, so the user can retrieve the annotation with his/her NFC reader. On the other hand, indirect access to the annotation means the user needs to use third party technology such as the Internet in order to get the annotation that was associated with the object. Direct and indirect access have advantages and disadvantages in the matters of speed, cost, the information size and other technical issues of accessing the annotation.

Definition 3.2 (accessibility)

We define $\boldsymbol{ACC}$ to be a set of accessibility types, and $\boldsymbol{acc} \in \boldsymbol{ACC}$ denotes a type that matches one of the $\boldsymbol{ACC}$ elements which are {direct, indirect}.

C. Noticeability

This property describes the means of discovering the available annotation at a certain place (or for a certain object). The user can either discover an annotation by his/her eyes without using any technology, e.g. if there is a sticker or an RFID tag associated with the physical entity. On the other hand, the annotation may be invisible and the user may need a mobile device to discover the annotations available around him/her. For example, if an object was tagged by using a GPS coordinate, the user needs a mobile with GPS in order to discover the annotation associated with those coordinates.

Definition 3.3 (noticeability)

We define $\boldsymbol{NOTC}$ to be a set of noticeability types and $\mathbf{notc} \in \mathbf{NOTC}$ as denoting a noticeability value. The set $\boldsymbol{NOTC}$ = {visible, invisible}.

D. Mapping

Mapping is a property that describes the relationship between the annotation and the annotated entity(-ies): one-to-one, which means one annotation is only associated with one entity, one-to-many, which means one annotation is associated with more than one entity at the same time (e.g., an annotation for a collection of furniture in a room), many-to-one, which means one entity is associated with many annotations. The mapping here can also be many-to-many, so that one annotation is associated with many entities and one of the entities also has many annotations.

Definition 3.4 (mapping)

We define $\boldsymbol{MP}$ to be a set of mapping types, namely, {one-to-one, one-to-many, many-to-one, many-to-many}, and $\mathbf{mp} \in \boldsymbol{MP}$ denotes a mapping type.

E. Context Dependency

This property describes all the possible conditions and situations that make the annotation readable and accessible to the users. The annotation could be *static*, which means it will be available every time and under any circumstances. However, the annotation could be *dynamic,* which will be available only under certain conditions and circumstances.

Therefore, context dependency will contain all the conditions that allow the annotation to be active and available to the users. An example of a dynamic annotation is when an annotation depends on many contexts such as location, time and weather conditions. So, all the three conditions must be met in order to be readable (or even visible) or active to the users. The context dependency also includes the dependency between the entities for annotation access – the idea is that an annotation can only be read when all entities are detected. When there is one annotation associated with many objects, some of the annotations could be a standalone annotation or others are dependent annotations. For example, the standalone annotation can be retrieved if the user reaches at least one object (i.e., able to read the NFC tag on the object), whereas, to read the dependent annotation, the user must get all objects (or read the NFC tags of the collection) at the same time (or within a fixed period). This situation is called *entity dependency*.

Another value of this property can be *nearby object*, or *nearby person*, where the annotation then becomes available only when there is another entity nearby such as a person passing the entity, i.e. the annotation for that entity (e.g., room) becomes available when the particular user is nearby. The value in this property can vary from one annotation to another and may have many values, so the following set of values for this property is just one possibility.

Definition 3.5 (context dependency)

We define $\boldsymbol{CX}$ as a set of contexts, and $\mathbf{cx} \in \mathbf{CX}$ is a context (or a type of context information).

$\boldsymbol{CX}$ = {location, time, date, nearby person, nearby object, entity dependency}.

F. Annotation cooperation

This property is concerned about the semantic interaction between the annotations, which we classify into collision and interference. In the case of interference and collision between annotations, there must be some mechanism to overcome any problems of this kind. The interference is the situation where there is an overlap between two annotated entities such as a floor and a room in that floor, and each one of them has its own annotation, and so, there is a need to combine the annotations (for the room); collision refers to the situation where there is more than one annotation for one entity and these annotations contradict (semantically) one another.

This property aims to organize the annotation, control their interactions, and give them different priorities. Some annotations may be allowed to be combined with others, whereas some annotations may make no sense when it is combined with others. Therefore, the annotation should contain some metadata that specifies how it might be used with other annotations. So, in the previous interference example, one of the annotations may be marked with higher priority in its metadata, so that when the interference/collision occurs, this annotation will have priority over the other annotations or it can just be combined (presented together for the user).

Definition 3.6 (annotation cooperation)

We define $\boldsymbol{ACOP}$ to be a set of annotation cooperation modes, and $\boldsymbol{acop} \in \boldsymbol{ACOP}$ = {priority, combination}.

G. Anchoring

This is about the technology that is used to mark the target. Such as 2D barcode, GPS coordinates RFID tag or image recognition.

Definition 3.7 (anchor)

We define $\boldsymbol{ANC}$ as a set that contains all anchoring types, and $\mathbf{anc} \in \boldsymbol{ANC}$ is a type of anchoring type.

After we defined the annotation and the target, we now give the definition of the link, between an annotation and a target.

Definition 3 (link)

A link is 7-tuple of the following form:

$$\boldsymbol{l} = < \boldsymbol{cx}, \boldsymbol{mp}, \boldsymbol{acop}, \boldsymbol{asso}, \boldsymbol{acc}, \boldsymbol{notc}, \boldsymbol{ancr} >$$

where $\boldsymbol{L}$ is a set of links, and $\boldsymbol{l} \in \boldsymbol{L}$ is a generic instance of the link,

$\boldsymbol{cx}$ is the annotation context dependency, i.e. $\boldsymbol{cx} \in \boldsymbol{CX}$,

$\boldsymbol{mp}$ is the mapping property which presents the relationship between the annotation and the entity, i.e. $\boldsymbol{mp} \in \boldsymbol{MP}$,

$\boldsymbol{acop}$ is the annotation cooperation mode, $\boldsymbol{acop} \in \boldsymbol{ACOP}$,

$\boldsymbol{asso}$ is the association type, i.e. $\boldsymbol{asso} \in \boldsymbol{ASSO}$,

$\boldsymbol{acc}$ is the accessibility type, i.e. $\boldsymbol{acc} \in \boldsymbol{ACC}$,

$\boldsymbol{notc}$ is the noticeability, i.e. $\boldsymbol{notc} \in \boldsymbol{NOTC}$, and

$\boldsymbol{ancr}$ is a generic anchor type, i.e. $\boldsymbol{ancr} \in \boldsymbol{ANCR}$.

Now, after defining the three parts of a physical annotation: annotation, target and link, we give the definition of the PA:

Definition 4 (physical annotation)

A Physical Annotation $\boldsymbol{pa}$ is 4- tuple of the form:

$$\mathbf{pa} = < \boldsymbol{paID}, \boldsymbol{ann}, \boldsymbol{t}, \boldsymbol{l} >$$

where $\boldsymbol{pa}$ is a generic physical annotation, i.e. $\boldsymbol{pa} \in \boldsymbol{PA}$,

$\boldsymbol{paID}$ is the unique number to identify each PA, i.e. $\boldsymbol{paID} \in \boldsymbol{PAID}$,

$\boldsymbol{ann}$ describes all the properties which belongs to the annotation itself (as in Definition 1), i.e. $\boldsymbol{ann} \in \boldsymbol{ANN}$,

$\boldsymbol{t}$ refers to the properties which describe the target, which is the annotated entity (as in Definition 2), i.e. $\boldsymbol{t} \in \boldsymbol{T}$,

$\boldsymbol{l}$ is the link between the annotation and the entity (as in Definition 3), i.e. $\boldsymbol{l} \in \boldsymbol{L}$, and we denote the set of physical annotations as $\boldsymbol{PA}$.

III. Analyzing Existing Annotation Systems

After providing the formal definition above, we now apply it to four popular Physical Annotation systems that have been used: Yellow Arrow, which is older, and current systems Wikitude, Layar and Junaio. The point is to validate our definition and to see how general the definition covers different aspects of the other systems' conception of PAs.

Yellow Arrow

Yellow arrow is one of the well-known PA systems. Yellow arrow was a great idea when it was introduced in 2004, and is still used in some cities till now [5]. The idea is to share information about a significant place. This information can be shared by using a yellow arrow sticker with a unique identifier pointing at the interesting place; the user can send an SMS text on finding a yellow arrow and then the server pings back the related information about that place. The annotation content doesn't have any formal format or structure. It was basic and textual. Based on our PA definition, the three tables below show how Yellow Arrow does not have a lot of interactive annotation features.

The annotation has a unique identifier. However, it doesn't have an annotation type, user, group or cooperation features. In the target side, the target doesn't have a unique number, so that when the user wants to add another annotation s/he has to add another yellow arrow with a unique number to annotate the same entity. Moreover the target entity doesn't have a target type. Also, the PA doesn't have context dependencies, mappings, or annotation cooperation features. Association is often strong. The accessibility is indirect; the user gets the information after he/she sends the server. Noticeability is such that the yellow arrow sticker is always visible and the yellow arrow sticker is the anchor.

Property	Value
Ann ID	Yes
Ann type	N/A
Media	Text, Voice
Content	Free format
Author	Yes
User	N/A
Group	N/A
Cooperation	N/A

Table 1: Yellow arrow annotation

Property	Value
Target ID	N/A
Target Type	N/A

Table 2: Yellow arrow target properties

Property	Value
Context dependency	N/A
Mapping	N/A
Annotation cooperation	N/A
Association	Strong
Accessibility	Direct
Noticeability	Visible
Anchor	Sticker

Table 3: Yellow arrow link properties

Wikitude

Wikitude is one of the first applications that provides an augmented reality browser for the real world, by using a mobile device which includes a built-in GPS, accelerometer, compass and camera. The user can access the information that annotates a physical object via the device. The annotations are rich-media and vary from simple text to complex multimedia. The annotations also are categorized into layers which are called *worlds*, so the user is able to choose one (or more) of them for viewing. However, the layers may be based on a third party provider who may not categorize the contents well. Examples of the third party provider are twitter, YouTube and Flickr. The layers can be authored by any organization or person to create a specific layer of certain type. In Wikitude, an annotation is not independent by itself. It is a part of the target properties, so it is stored in the database accessed via the target's IDs. The annotation content doesn't have any formal format; the user is free to write any annotation in any format without any limitations. There is user identification in the system; however, there is no group or user authentication to access the annotations. So any layer or annotation can be accessed by any user without any constraint. Therefore, the cooperation feature is not implemented in the system, which means the author cannot choose the target user in order to access his/her layer. Target type as well is not well defined here, with reference to our PA formalization. So any annotation can annotate any entity without constraints. The target ID is the location coordinates, but this may not be adequate, especially when the user wants to annotate several different entities at the same location. Regarding the PA link between the annotation and the target, the context dependency is often user location only. No further context is used in the system. The mapping feature is not well implemented in the system, e.g., the annotation cannot be applied to more than one physical entity. The relationship between the annotation and the target is often only one-to-one. Moreover, the annotation cooperation feature is not explicitly supported. So the annotation could be a strong annotation for the target, or just a weak annotation. Accessibility in Wikitude is indirect, the user needs Internet connectivity to access the related information. Moreover, since the system is using augmented reality technology, the noticeability here is "invisible" and the annotation can only be noticed/discovered using an appropriate device which must have the required capabilities.

Property	Value
Ann ID	N/A
Ann type	Yes
Media	Mix
Content	Free format
Author	Yes
User	Yes
Group	N/A
Cooperation	N/A

Table 4: Wikitude annotation properties

Property	Value
Target ID	Location coordinates
Target Type	N/A

Table 5: Wikitude target properties

Property	Value
Context dependency	Location
Mapping	N/A
Annotation cooperation	N/A
Association	Both
Accessibility	Indirect
Noticeability	Invisible
Anchor	GPS, accelerometer, compass

Table 6: Wikitude link properties

Layar

Layar[6] is a famous example of an augmented reality physical annotation system. As far as we can observe, similar to Wikitude, the annotation is not independent by itself, it is part of the target properties. Layar has the target type property such as education type and so on. Similar to Wikitude, Layar also categorizes the annotations into *layers*, so the user can choose his/her preferred type of POIs, or annotation types.

Similar to the Wikitude's worlds, the layers may depend on third party content provider, so that sometimes the annotations may not be categorized properly into annotation types. The media property in Layar could be mix of different types such as video and text. The Content does not have any formal style, and so, the author can create annotations without any structure or constraint. In the last version of Layar (3.0)[7], a private layer can be created by using the feature of user authentication which will help the author to create groups which are given permissions to access the layer. For the target section in Layar, the target here is identified by its location coordinates, but this might cause problems when the user want to associate annotations directly with different entities at the same location. The target annotation depends on the location coordinates and its orientation. Moreover, the targets do not seem to be categorized into types. For the PA link, context dependency depends on the location only. There is no variation in mapping, which means the relationship between the annotation and the target is, by default, one-to-one. Also, there is no annotation cooperation. The annotation association is only indirect which mean the user needs Internet connectivity to access the annotation. For Noticeability, the annotation is always invisible, and only can be seen by using a mobile device. The annotation is anchored using GPS, accelerometer and compass.

Property	Value
Ann ID	N/A
Ann type	Yes
Media	Text, image. Sound, video, 3D models
Content	Free format
Author	Yes
User	Yes
Group	Yes
Cooperation	Yes

Table 7: Layar annotation properties

Property	Value
Target ID	Location coordinates
Target Type	N/A

Table 8: Layar target properties

Property	Value
Context dependency	Location
Mapping	N/A
Annotation cooperation	N/A
Association	Both
Accessibility	Indirect
Noticeability	Invisible
Anchor	GPS, accelerometer, compass

Table 9: Layar link properties

Junaio

Junaio [8], created by the German company Metaio GmbH, is another famous example of the current Augmented Reality browser. It has many features such as supporting of location based services outdoor and indoor. It also supports marker and marker-less image recognition. A Junaio annotation, similar to the Layar system, is part of the target properties. The user can categorize the content and the annotations into categories called *channels*, so the user can chose his/her preferred channels such as game, or travel. The channel can be provided by a third party content provider such as Wikipedia, or Twitter. The media used is rich and could be text, image, sound, video, or even 3D models. Also, the content can be in any form without any structure or rules. The author can create a public channel for everyone, or s/he may make it private and share it with his/her friend only. For the Target section, target in Junaio has an identifier which is called POI ID of interaction; however, the target doesn't have typing. For the PA link section, context dependency is based on the position of the user only. There is no mapping technique used in the system, which means the relationship between the annotation and the target is one-to-one only. Between the annotations there is no annotation cooperation which means there is no interaction between the annotations themselves. The annotation association is only indirect which mean the user needs Internet connectivity to access the annotation. The annotation is invisible and visible, invisible when the GPS and compass are used or visible when the LLA markers are used (a LLA marker encodes the latitude, longitude and altitude). The annotation anchor is using GPS, accelerometer, compass and image recognition, or 2D barcodes (LLA). In Junaio, users can create two types of channels, the first one is the location based information channel, which can be used outdoors by using a GPS, or indoors by using the LLA marker. The second type of channel can be a Glue channel which is used for optical tracking via image recognition.

Property	Value
Ann ID	N/A
Ann type	Yes, Channels
Media	Text, image. Sound, video, 3D models
Content	free format
Author	Yes
User	Yes
Group	Yes
Cooperation	Yes Private, shared , public

Table 10: Junaio annotation properties

Property	Value
Target ID	POI ID
Target Type	N/A

Table 11: Junaio Target properties

Property	Value
Context dependency	Location
Mapping	N/A
Annotation cooperation	N/A
Association	Both
Accessibility	Indirect
Noticeability	Invisible / visible
Anchor	GPS, accelerometer, compass, image recognition , 2D barcode

Table 12: Junaio link properties

Discussion. From the analysis above, we analyzed four systems to study the features of PA systems, and also the strength and the weakness of the currently used systems. As we discussed in the analysis, the current systems still have many drawbacks such as the lack of the mapping property, limited context dependencies and many other properties. The main problem between all systems is that, there is no clear definition to define the PA, and the three components of the annotation system which are the annotation, the target and the link. When we studied the annotation systems over the last ten years, we found there is a need to provide a formal definition for the Physical Annotation which is general to cover all aspects of the PA, and amenable to systematic extensions.

IV. Generic Reference Architecture

In this section, we propose a generic reference architecture for physical annotation systems based on our three-part formal definition of PAs.

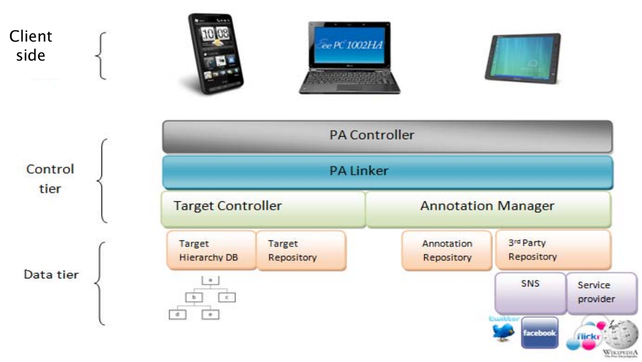

Fig.1: Generic reference architecture for PA systems.

Client side: this is a user handset. By using a mobile phone, a laptop or a tablet, users can access, retrieve and modify a physical annotation.

PA Controller: this tier controls the PAs in general; after the system retrieves the annotation from the lower tiers, it manages the interference/collision (if any), and all possible situations that may affect the annotation.

PA Linker: the main job of this layer is to map annotations to entities/targets. This tier includes conditions of the linking, and uses the PA link properties such as context dependencies and mapping. It manages entity containers/collections when there are more than one entity annotated with the same annotation (an annotation for a collection).

Target Controller is responsible for the annotated entities. This tier includes two components:

- *Target Hierarchy Nodes DB:* tracks the entity in its spatial hierarchy (e.g., within a room on a floor in a building), giving the user the option to retrieve the parents node
- s annotations as well. For example, if the user in room A on the 2nd floor in building B, we can present this in a spatial tree. The user may like to retrieve the annotations for that room and the floor as well. So this tier will locate the entity in its hierarchy and give the user the option to retrieve the annotations belonging to the space containing the location, not just the annotations for the spot/location.
- *Target Repository:* contains information about the annotated entity, whether it is a small object, location or person. The repository includes also the entity's target type which will help to determine the possible associated annotations of this object.

- *Annotation Manager*: this layer is about the annotation content which is the description of the annotated entity. The contents here are stored in two repositories:
- *Annotation Repository*: which is the system's own annotations.
- *External 3rd party annotation repositories*: the annotation content can be provided by third party providers such as social network services, e.g., Facebook can provide some useful annotations, and Wikipedia, so this layer refers to such information services.

Scenario. To illustrate our architecture, we apply it to a shopping centre such as the Northland shopping centre in Melbourne; the centre contains shops, each shop has sections, and each section has items. The Target Hierarchy DB stores the structure of the entities of the centre in a DB in a tree form. The Target Repository stores all annotated entities in the centre. In the system, as we drill down in the hierarchy; there are different annotations for different levels of the tree, corresponding to the structure of the shopping centre. For example, there are annotations that apply to the whole centre and can be retrieved from anywhere in the centre. Then for each shop, there are special annotations for that shop, in general such as "Kmart, has 30% off electronic devices". Then, inside a shop, there are further annotations for items on promotion, such as "buy this item, and get one free". Or the annotation could contain more description about that item. The role of the Target Controller is to link the entity in the Target Repository to its spatial hierarchical information, which gives the users the chance to retrieve the annotation for that entity, and also annotations for the spaces containing the entity. In the annotation section, Annotation Repository stores all annotations; this component can be a third party annotation provider, such as YouTube, or can be a private annotation repository designed for a particular place. The Annotation Manager manages the annotation retrieval from different repositories. PA linker's role is to link an annotation to a particular target entity. This includes all the linking conditions, mapping and PA link properties that we discussed in previous sections. We assumed in our model that annotating an entity can be via dynamic annotations, which means the annotation is only valid in certain contexts (e.g., for a particular time only). This assumption makes the system more efficient and useful; the PA Controller manages this dynamic process. For example, if we have two different annotations for a particular entity, these two annotations can be applied together, and sometimes, if there is a collision or interference between them, the system manages this. To explain this further, let us assume that certain bread in the Safeway Supermarket is annotated with 20% discount on Mondays. But another annotation says there is only 10% off the bread after 6pm everyday because it's not fresh. In this case, there is collision between these two annotations on Monday after 6pm. So, this tier manages such cases, by allowing the two annotations to be combined (e.g., give the consumer a 30% discount, or by giving one annotation priority over the other one).

We now provide a case study based on the scenario above which provides grounding for our formal model for this scenario. Assume we have the following *physical annotation* for a given coffee shop "30% off on coffees from 3pm to 5pm". Let us apply our formal definition to this annotation. We have the three main parts of the physical annotation as we discussed earlier: the annotation, the target and the link.

The annotation is an 8-tuple of the following form

$$\boldsymbol{ann} = <\boldsymbol{aID}, \mathbf{at}, \mathbf{md}, \mathbf{ac}, \mathbf{au}, \mathbf{user}, \mathbf{gr}, \mathbf{cop}>$$

where: $\boldsymbol{aID}$ is an annotation identifier = "an4412",

$\boldsymbol{at}$ is the annotation type = "commercial",

$\boldsymbol{md}$ is the media type = "text",

$\boldsymbol{ac}$ is the annotation content = "30% off on coffees from 3pm to 5pm", $\boldsymbol{au}$ is the annotation's author = "the shop's manager", $\boldsymbol{user}$ is the annotation user = "general public",

$\boldsymbol{gr}$ is the user group = {general public, shop's staff}, and

$\boldsymbol{cop}$ is the user cooperation type = "public".

Therefore, the annotation here, $\boldsymbol{ann}$ = <"an4412"," commercial", "text" , "30% off on coffees from 3pm to 5pm", "the shop's manager", "general public", "general public, shop's staff" , "public" >.

The second part is the annotated entity, i.e. the target, which is a pair as follows: $\boldsymbol{t} = <\boldsymbol{t\_id}, \boldsymbol{t\_t}>$

where: $\boldsymbol{t\_id}$ is the target ID = "TL512", $\boldsymbol{t\_t}$ is the annotated entity type or target type = "location" (since this is a shop).

Therefore, the target $\boldsymbol{t} = <\textbf{"TL512"}, \textbf{"location"}>$.

The third part is the PA link which is a 7-tuple of the form: $\boldsymbol{l} = <\boldsymbol{cx}, \boldsymbol{mp}, \boldsymbol{acop}, \boldsymbol{asso}, \boldsymbol{acc}, \boldsymbol{notc}, \boldsymbol{ancr}>$

where: $\boldsymbol{cx}$ is the annotation context dependency = {"shop's location", "time 3pm-5pm"},

$\boldsymbol{mp}$ is the mapping property = "one-to-one",

$\boldsymbol{acop}$ is the annotation cooperation = {"priority = 1","combination = N/A"} which means this annotation can't be combined with other annotations if there is an interference with other annotations,

$\boldsymbol{asso}$ is the association with the target = "strong",

$\boldsymbol{acc}$ is the accessibility = "direct", because we assume the user can get the annotation (small enough to fit) from the RFID/NFC tag without needing to access the Internet,

$\boldsymbol{notc}$ is the noticeability = "visible", because there is an RFID tag visible to the human eye, and

$\boldsymbol{ancr}$ is the anchor type = "RFID tag".

Therefore, the PA link = <{"shop's location", "time 3pm-5pm"}, "one to one", {"priority = 1","combination = N/A"}, "strong"," direct"," visible"," RFID tag"> .

And now, the physical annotation is, hence, a 4-tuple of the form: $\mathbf{pa} = <\boldsymbol{paID}, \boldsymbol{ann}, \boldsymbol{t}, \boldsymbol{l}>$, where:

$\boldsymbol{paID}$ is the unique number to identify each $\mathbf{PA}$ = "PA10077", $\boldsymbol{ann}$ is the 8-tuple annotation part as above,

$\boldsymbol{t}$ is the 7-tuple target part, and $\boldsymbol{l}$ is the link part as above.

V. RELATED WORK

For a decade, the Physical Annotation (PA) has been an active area of research. The main idea is to annotate physical objects with digital information and share it with others. Much of the research in this field focused on the technologies that are used for the annotation such as the marker in yellow arrow, RFID tags in the Cooltown project [9], or more recently, augmented reality (AR) in Wikitude [10] and Layar [6]. Some of the other researchers have focused on the markup languages used to present the contents of the annotations such as ARML [11] and Junaio XML [8]. There has also been work done on formalizing the annotation in digital libraries/contents in general, such as [12]. However, as far as we know, there is little research done on the formalization of the PAs. Therefore, this paper aims to provide a formal model for the PA. The paper presents the properties of PAs in order to provide a potential standard for all applications which are based on this principle, thereby enabling interoperability, to consolidate developments in the area and as a step towards more complex PA systems, analyzable formally.

VI. CONCLUSIONS

In this paper we provided a formal definition for physical annotations (or PAs) and outlined a generic reference architecture based on this model. We also analyzed four of common systems that are used for physical annotations according to our PA definition. As we discuss these systems, we found that there is a need to provide a standard for the physical annotation systems in terms of the definition and the range of possible features. The future work of this paper is to refine our model, and provide a markup language that can be a standard for PA systems which can cover more features and aspects of PA.

REFERENCES

1. Hansen, F.A. Ubiquitous annotation systems: technologies and challenges. in HYPERTEXT '06: Proceedings of the 17th ACM conference on Hypertext and hypermedia. 2006.
2. Wither, J., DiVerdi, S., and Höllerer, T., Annotation in outdoor augmented reality. Comp. & Graphics, 2009. 33(6): p. 679-689.
3. Weigelt, K., Hambsch, M., Karacs, G., Zillger, T., and Hubler, A.C., Labeling the World: Tagging Mass Products with Printing Processes. IEEE Perva. Comp., 2010. 9(2): p. 59-63.
4. Tanabe, J.P. Prehistoric Cave Paintings. Available from: http://www.squidoo.com/cavepaintings (accessed 6 July 2009).
5. YELLOW ARROW 2004; Available from: http://yellowarrow.net/v3/index.php (accessed 8 June 2009).
6. http://layar.com/ (accessd 1 September 2010).
7. http://layar.pbworks.com/w/page/7783257/User-authentication-in-a-layer , accessed 12 March 2011.
8. junaio. http://www.junaio.com/publisher/gettingstarted.
9. Kindberg, T., Barton, J., Morgan, J., Becker, G., Caswell, D., Debaty, P., Gopal, G., Frid, M., Krishnan, V., and Morris, H., People, places, things: Web presence for the real world. Mobile Networks and Applications, 2002. 7(5): p. 365-376.
10. Wikitude World Browser Available from: http://www.wikitude.org/ (accessd 1 September 2010).
11. Martin Lechner and Tripp, M., ARML - An Augmented Reality Standard, in Mobile AR Summit MWC 2010. 2010.
12. Agosti, M. and Ferro, N., A formal model of annotations of digital content. ACM Transactions on Info. Sys., 2007. 26(1).