# Classification of Spontaneous Device Association from a Usability Perspective

Ming Ki Chong, Hans Gellersen
Computer Department, Lancaster University, UK
{chong, hwg}@comp.lancs.ac.uk

## ABSTRACT
As computing devices become ubiquitous, devices are expected to encounter and associate with one another spontaneously to form ad hoc networks for sharing resources. One of the challenges in research remains in the task of making a device association scheme secure and, at the same time, easy to use. There have been numerous proposed solutions from literatures, with each solution designed for specific purposes and scenarios. At present, there is no clarity of differences and similarities amongst those association methods. In this paper, we present a classification of device association methods based on categories that influence the usability of an association.

## Categories and Subject Descriptors
D.4.6 [**Operating Systems**]: Security and Protection – Authentication; H.1.2 [**Information Systems**]: User/Machine Systems – Human factors.

## General Terms
Security, Human Factors.

## Keywords
Classification, Spontaneous association

## 1. INTRODUCTION
Short-range wireless communication is becoming popular in electronic devices; the communication capability has made many new services possible. For example, standards like Bluetooth, Wi-Fi, etc. have enabled devices to form ad hoc networks to share resources and information. Nevertheless, before devices can exchange data, they need to execute an *association* process for introducing and forming a connection amongst themselves with one another. The process involves the new connecting device(s) finding the targets to establish a communication channel. However, unprotected communications are vulnerable to interception; due to the inherent open broadcast nature of wireless communication, wireless channels require security. To establish a *secure connection*, a common secret key must be agreed among the participating devices during their association.

At present, the typical device association model[1] is based on a 3-step process: *identification* of neighbouring devices (e.g. Bluetooth discovery protocol), user *selection* of the target device(s), and *authentication* by a passkey (usually in alphanumerical form). However, this model has both usability and security problems. For instance, a long and random passkey can increase security but, simultaneously, it decreases usability; conversely, using a short passkey is vulnerable to passive attacks. As technology evolves, some devices may not even have an input interface that supports passkey entry; thus, in reality, it is impractical to mandate a single association model for all kinds of devices [33]. To address this concern, many alternative solutions have been proposed in research. Each of the proposed solutions targets a specific scenario with different sets of hardware, requirements and limitations; in other words, different association models have been suggested for their individual purposes. To further understand the overall complexity, we need an overview of the current situation; a general classification of device association models is needed. In [33], Suomalainen *et al.* present a systematic taxonomy of security protocols for key establishment; while their focus was on security, in this paper, we present a classification of spontaneous device association from a usability perspective. The goal of this classification is to identify categories that influence the usability of an association, and the categories serve to help designers and researchers to consider the contextual factors of an association.

[1] Suomalainen *et al.* define an *association model* as the part of the association procedure that is visible to the user [33].

## 2. CLASSIFICATION CATEGORIES
Our focus in this paper is oriented from a non-technical viewpoint; we present a classification based on usability aspects. We identify and define categories (and their sub-categories) for classifying the components of a device association. We refer readers to figure 1 for an overview of the classification.

At the highest level (the root), we have *Spontaneous Device Association*. The aims of a spontaneous association are to achieve good usability and good security. An association method requires fast and intuitive interaction where users spend minimal time to form a device connection and simultaneously the method must maintain and support maximal protection to prevent harmful attacks; in other words, minimum effort and maximum security.

Below the top level, we define three main criteria: *Technology*, *User Interaction* and *Application Context*. The technology category focuses on the physical aspects of the hardware equipment and the communication channel. The user interaction criterion refers to the relations between users and hardware based on the users' action. And lastly, application context refers to the circumstances happening during an association. In the rest of this paper, each criterion and it sub-levels are discussed with definitions and examples.

### 2.1 Technology
Any device with a network capability is ideally able to associate with other devices that have the same network capability. However, designers need to consider the physical and technical limitations of the devices to design a usable association. For in-

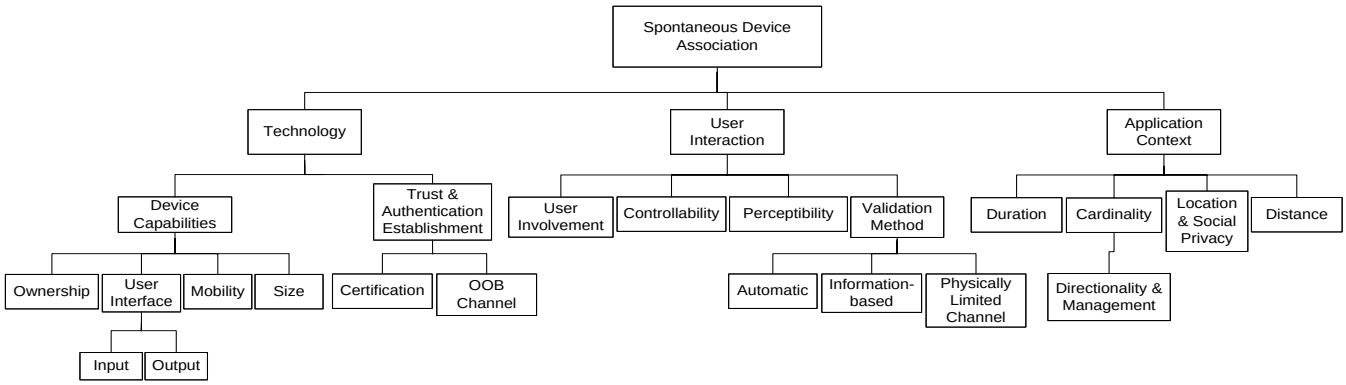


**Figure 1. Overview of categories of spontaneous device association from a usability perspective**

stance, a heavy and bulky device or a device without motion tracking capability cannot adopt the shaking method to establish a connection [14,20]. To distinguish the limitations, we categorise two factors that influence the design of an association model: *Device Capabilities* and *Trust & Authentication Establishment*.

### *2.1.1 Device Capabilities*

In short, the device capabilities category refers to the characteristics of the physical devices. This category can further be subdivided into the following variables:

**Ownership**: We classify three types of device ownership: *Personal* (a device that solely belongs to one person), *Private Group* (a device that is shared by a private group of users), and *Public* (a device that is accessible to anyone). A public device has no privacy as anyone can access it (like a public library computer); conversely, a personal device is fully private thus personal information is stored. The ownership of the device directly affects users' willingness to associate the device with other foreign devices. For example, users are more reluctant to connect their personal device to a public terminal, as the users are less certain about the security and trustworthiness of the public device; for such type of association, discreetness and robust security are needed.

**Required User Interface**: This category refers to the hardware features that facilitate an interaction between users and their devices. The category comprises of two classes: *Input* and *Output*. The input attribute refers to the features that allow the users to manipulate a system, whereas the output attribute refers to the features that allow the system to indicate the effects of the users' manipulation. Different devices have different input and output facilities. Identifying a full set of all possible hardware features is impossible. Instead, the purpose of this category is for identifying requirements of an association model. When designing an association, designers should consider the required input and output features and how they can affect the overall usability. For example, shaking requires a motion sensor on each device; audio requires devices with an audio emitter (speaker) and an audio receiver (microphone); and so forth.

**Mobility**: Five types of device mobility are identified: *Mobile*, *Portable*, *Situated*, *Wearable* and *Automobile*. For simplicity, we use examples to illustrate the five categories. A cellular phone is mobile, as it is light-weighted and easy to carry. A laptop computer is portable, as a user can carry it while moving, but the bulkiness limits its mobility. A cinematic projector is situated, as the device is anchored in a fixed location. However, wearable devices do not fit into any of those three categories. Wearable devices are neither mobile nor portable, as they must be placed in a fixed location of the wearer to function whilst, at the same time, the devices are not situated as the wearer is mobile. For example, the Nike+iPod Sports Kit [1] sensor is wearable as it is embedded inside a shoe, yet it is not mobile as it must be on a specific location of the wearer to capture information. Finally, a standard vehicle nowadays has many embedded devices; the devices are fixed onto the vehicle to support many functions that offered by the vehicle as a whole. Thus, we can view an automobile as a single unit device.

**Size**: Another physical factor that influences the usability of an association model is the size of the connecting devices. Fast moving interaction techniques are mostly suitable for small devices; however, as we understand from Fitts's Law [9], smaller objects are also harder to aim/target for interaction that require precision, like pointing a laser at a receiver to authenticate devices [16,22]. An association model that is suitable for one type of device may not be suitable for another. We therefore classify devices according to their dimension to recognise their differences. However, we cannot classify devices based on their exact measurements; instead, we use metaphors to define their different sizes: *Candy/Pin* (e.g. small microphones, body sensors), *Soap* (e.g. mobile phones, digital audio players), *Book* (e.g. laptops, tablet PCs), *Furniture* (e.g. desktop PC, tabletop computer), *Notice Board* (e.g. large display monitors), *Wall* (e.g. electronic billboards; similar to Notice Board but much larger in size, such that it cannot fit indoors) and *Irregular* (everything else).

Device mobility is often seen as related to the size and the weight of the device, e.g. mobile devices are generally small and lightweight. However, this assumption is not strictly true. A device may be small in size but attached to a fixed location, thus it is not mobile, but situated instead. In other words, the two categories, mobility and size, are loosely related with exceptional cases.

### *2.1.2 Trust and Authentication Establishment*

This category may refer to many aspects in security, including protocols and algorithms. Here, we are particularly interested in aspects that are related to usability; we therefore neglect anything

that does not involve the users. In addition, this category refers to the mechanisms that enable devices to authenticate communication channels and to establish trust between devices.

Before designers select an authentication technology, they need to acknowledge that not all associations require security. The need of security depends on the application context, there are scenarios where security can be omitted (we discuss this in detail later). Authentication may be optional if the context of the application is risk-free. Nevertheless, in spontaneous interaction where communications happen unexpectedly, it is likely that devices encounter in an unpredicted location and associate in a previously untrusted manner; in which case, security is required. Here, we identify two approaches that can be used for establishing trust and authenticating devices during an association.

**Certification**: Digital certificate can be used for establishing trusted and secure channels between devices. The use of certificates requires the associating devices to recognise an external mutual trusted party (a certificate authority) that signs their certificates. However, this is an unattractive option as it requires a huge infrastructure which is expensive [19]. Furthermore, certificates are used for verifying a binding between a public key and an identity in the form of a digital address. Yet, for device association, such binding is not sufficient, as we still need a binding between a public key and the target device itself [17]. As a result, a user is still required to validate the fingerprints of the public keys to ensure the keys are affiliated with the correct devices. Although certification is impractical for large-scale implementation, it is still usable in small-scale if a common trusted authority amongst the devices is known prior the association.

**Out-of-band (OOB) Channel**: An alternative approach is via the use of a secure OOB channel. While wireless messages are transmitted over an insecure channel (the *in-band channel*), authentication data can be transmitted over an external channel (the *out-of-band channel*). An OOB channel is established with the aid of a human operator. The user initiates a device association by accomplishing a simple task like entering a text into the selected devices; as a result, the communication is implicitly authenticated by the task, as no adversary can forge the user's action. Nonetheless, due to the involvement of a human user, an association model must only adopt non-laborious tasks to achieve good usability, and it must also be inaccessible by assailants to maintain high security.

A wide range of OOB channels have been suggested from research. Here, we briefly identify the channels as follow: *Physical Contact* e.g. electrical contact [32] and human touch contact (tactile) [24]; *Computer Vision* e.g. using a camera to capture a visual code of the target device [23] or to capture blinking patterns [28,29]; *Light Beams* e.g. infrared beams [2,34] and laser beams [4,16,22]; *Sound Waves* e.g. acoustics audio [11,12,30] and ultrasound [17,21]; *Human Actions* e.g. synchronous button presses [15,26,31], random motions (or shaking) [5,14,20], synchronous gestures [13], gestures entry [25], passkey entry [6,10], text comparison [6,10,18,36] and biometrics [7]; *Distance Proximity* e.g. near field communication [27]; and *Radio Signals* e.g. radio environment comparison [35] or differences in broadcast packets [8].

The selection of OOB channels is not exclusively limited to using only one OOB channel. Multiple OOB channels can be used simultaneously. For instance, the Loud & Clear [11,12] concept adopts both audio and human channels; it involves a user comparing audio vocalisations to authenticate devices.

## 2.2 User Interaction

To understand the usability of an association model, we evaluate the interactions that users must perform; thus, we need to consider factors that influence the interaction interface of an association model. In this section, we define and discuss the following four categories: *User Involvement*, *Controllability*, *Perceptibility* and *Validation Method*.

### *2.2.1 User Involvement*

Using the categorisation of different OOB channels, we can further classify them based on users' role during an association.

Different OOB channels require different involvement of the users. Some channels require minimum user involvement, e.g. a user initiates the association and the rest of procedure executes automatically. We refer this type of user involvement as *Device-to-Device* (D2D) (the terminology is adopted from [29]) because the authentication data is generated and transferred directly from one device to the receiving devices. For example, the techniques of pointing a light beam to a photo-sensor [2,4,16,22,34] are D2D, since the user is only required to point the beam and the rest is completed by the devices.

The second type of user involvement is a user generates and inputs authentication data directly into the associating devices and the data is used by the devices to authentication one another. We refer this as *Human-to-Device* (H2D). For example, the technique of shaking to associate devices [5,20] is H2D, as it requires a user to input the authentication data (captured from motion sensors) directly into all the devices. H2D requires more users' involvement than D2D as H2D require the users to generate the authentication data. From users' viewpoint, it is possible that H2D is more intuitive as the users' involvement allows them to participate in the process; thus, they are more aware of the authentication being taking place. However, since authentication data is generated by a human user, the data is predictable by examining the user's habit.

The third type of user involvement is having the users as the middle persons to convey authentication data. This requires a device first generating its authentication data and presents the data to a human user, and then the user inputs the data into the target devices. We refer this as *Device-to-Human-to-Device* (D2H2D), as it requires the authentication data to flow through two channels: first from device-to-human (D2H) and then human-to-device (H2D). An example of a D2H2D technique is the text comparison scheme [6,10,18,36]. The associating devices first negotiate a secure connection. After the negotiation, each of the devices displays a digital fingerprint (or a hash value) to its user. The user reads and compares the fingerprints: if they are identical, the user confirms and accepts the connection, else the user rejects it. Unfortunately, D2H2D schemes have 2 downfalls: (i) the representation of the authentication data is limited to a human interpretable form; (ii) the scheme is mentally demanding as users are required to examine the authentication data from every device. The mental demand increases as the number of devices increases. On the other hand, this scheme is suitable for devices with limited I/O capabilities, e.g. devices with a simple display and a binary input for accept or reject.

### 2.2.2 Controllability

In any association models, we can classify them based on how users control the connecting devices. We identify two categories: *Single User Control* and *Individual User Control*. Single user control refers to techniques that limit one person from a group of users to control and operate the entire association. For instance, the technique of shaking devices [5,14,20] is controlled by a single user, as only one person can shake the devices at once. If each individual user shakes their own devices, the devices would have different shaking patterns. On the contrary, individual user control techniques allow each user to operate their devices during an association.

In many cases, a user often wants to connect multiple personal devices e.g. connecting a mobile phone and a Bluetooth headset. In such a scenario, the user can operate the devices by oneself; thus, single user control techniques are appropriate. On the other hand, there are cases where a user associates his/her personal device to other peoples' devices; in such a scenario, each person should operate their own devices; thus, individual user control techniques are more suitable. From a social point of view, single user control techniques can be deemed as inappropriate in group associations. For example, personal devices (like mobile phones and PDAs) often contain private and sensitive information such as emails and personal pictures; some single user control techniques require users to surrender their physical possession of their devices to the person who conducts the association. This may be acceptable if the group of users socially trust one another; however, when strangers are involved, single user control techniques can raise security and privacy concerns.

Some association techniques can be classified as both single user and individual user; it depends how the techniques are executed. The text comparison techniques, for example, allow a group of users to select one member to perform the comparison task (single user) or, alternatively, all of the members can verbalize their authentication text message and perform the comparison task (individual user).

### 2.2.3 Perceptibility

Regular users do not understand the technical aspects of how an association is executed; instead, their understandings come from the action they need to perform. As a result, people interpret an association based on what they can gather from their sensory system.

Physiologically, humans have five natural input senses. So far, sight, hearing and touch are commonly applied in computer interaction. By exploiting those senses, designers can create different interfaces for users to associate devices (we found no research that explores taste or smell for device association). People can use their senses to construct a mental perception of the physical interaction of operating an association. By constructing a perception, the users can estimate the execution of the association. We therefore define perceptibility as the way how users perceive the interaction of an association model.

Every association has a two-part procedure: initiation and execution. The initiation must always be administered by a human user, as the user must interact with his/her device to start it; thus, the initiation stage is always perceived as interactive. The second stage, the execution of the association, is where we can identify different perceptibility.

We classify three types of perceptibility: *Tangible*, *Sensible* (or *Perceivable*) and *Non-interactive*. A tangible association requires its users to physically touch to interact with the connecting devices to accomplish the execution. For example, shaking requires a user to physically touch the devices; thus, the action is tactile and tangible. Conversely, the execution of a sensible/perceivable association can be perceived by seeing (e.g. a laser beam) or hearing (e.g. acoustics audio), but the execution of the association does not require users to interact with the devices physically. Lastly, a non-interactive association executes its procedure without the intervention of its users. For example, Amigo [35] uses radio signals which cannot be touched, seen nor heard; hence, from a user's viewpoint, the execution happens automatically.

### 2.2.4 Validation Method

Earlier, different OOB channels for device authentication were identified. From a user's perspective, the use of an OOB channel is deemed as a tool to transmit authentication data. To understand its usability, we need to analyse the channel from the interaction perspective. Users see the interaction as a way of validating an association; thus, the adoption of an OOB channel is a validation method in users' view. The validation methods that have so far been suggested in literatures can be described into three different approaches: *Information-based*, via a *Physically limited channel*, or *Automatic*.

A method that requires a human to authenticate devices via a piece of information, such as text, is classified as information-based validation. Two types of information-based validation were identified by Mayrhofer and Gellersen [20]: *human verification based on direct output of the associating devices*, or *direct user input of authentication material into the involved devices*. The former can be thought of as an interactive challenge-response protocol, with the user in the role of verifying device responses, while the latter as having the user in the role of providing explicit input into the involved devices to ensure that the intended devices authenticate.

Alternatively, a physically limited channel can be used, where the user is in the role of physically controlling the intended devices over a limited channel. Three types of limited channels are identified: *Location-limited*, *Movement-limited*, and *Time-limited*. A location-limited channel has the property that human operators can precisely control which devices are communicating with one another within a spatially-limited area [3]; for example, infrared and sound as they have a limited range. Defined by Mayrhofer and Gellersen, a channel is movement-limited, if it affords precise user control over which devices can communicate, by way of controlling their movement and by using movement as shared secret [20]. And, although there is no formal definition, we consider a channel is time-limited if a set of actions is executed by a user within a precise time interval. For example, SyncTap [26] allows a user to establish device connections through synchronous button operations.

Lastly, an authentication is automatic if it requires no user intervention. For example, authentication via a third party certificate or using radio features [35]. These methods can execute without the assistance of a human user.

## 2.3 Application Context

From a security standpoint, device authentication is always necessary as information that travels over any wireless channel is susceptible and vulnerable to attacks. In contrast, from a usability viewpoint, in some (or many) situations, security and authentication may seem unnecessary or paranoid. Consider the following instance: John is at home browsing pictures on his computer, he finds a few interesting pictures and he wants to upload them onto his personal device. In this scenario, John is at home, in a safe haven, and based on his surrounding context, his action is risk-free. So, is it necessary to inconvenience John for additional security? Some may argue that John's neighbours could tap into his connection; then the question becomes, is it worth compromising security for convenience.

When designing an association model, it is important to consider the context of the application to decide whether security is necessary. Security procedures may sometimes be annoying if users see no risk in the task; thus, security is best avoided if possible. To aid understanding the context of an application, we identify four categories: *Duration*, *Cardinality*, *Location Privacy* and *Distance*.

### *2.3.1 Duration*

Duration refers to the period of existence of a connection. For short-lived connections, like sharing an image file, users may see security as unnecessary if the authentication procedure lasts longer than the connection. The duration of a connection is predictable based on the objective of the connection. By estimating the duration, designers can decide if an association is suitable for the objective of the connection.

We define three types of duration: *Transient*, *Session-based* and *Permanent*. A transient connection terminates after a predefined condition is met and the duration expires within a certain time period that is calculable by examining the action. For example, file transfer is transient, as the connection terminates after the receiver has acquired the files and its duration is calculable by examining the size of the files and data speed of the channel. A session-based connection terminates after a predefined condition is met but the duration is not calculable. For example, network games and teleconferences are session-based. Their connections terminate once their sessions are over, yet the durations of the sessions are not calculable. A permanent connection ideally lasts indefinitely and it has no predictable terminating condition. The only terminating conditions are physical failure or a forced termination by a human operator. For example, the connections for wireless accessory devices like Bluetooth mice and keyboards are considered permanent.

### *2.3.2 Cardinality*

Most proposed methods in research are based on the idea of associating two devices (i.e. *pairing*), but, in reality, we often find situations where more than a pair of devices are involved. For example, a group of gamers want to form an ad-hoc network to play a network game. Association models differ as the number of devices changes. As more devices are involved, more user interaction is required. Adopting an association model designed for the wrong cardinality may negatively impact its usability. For example, the model of pointing a laser for device authentication can only execute a pair of devices at a time; using such scheme for multiple devices requires many interactions, and as the number of interactions increases, mental demand also increases and efficiency decreases.

Three types of cardinality for device association are identified: *Pairing*, *Physically Limited* and *Unlimited*. Pairing, as the name suggested, is strictly limited to two devices. Physically limited cardinality association models have the property of allowing many devices to authenticate at once; however, due to some physical constraints, the association can only accommodate a limited number of devices. For example, shaking to associate devices [14,20] is physically limited in cardinality, as the number of devices a user can hold with his/her hands is limited. In such a case, the cardinality depends on the physical sizes of the devices and the user's hands. Unlimited cardinality is theoretically not bounded by its physical factors. For example, passkey entry and text comparison have no upper limit. The number of devices does not create a limitation as long as the passkey or the text value can be acknowledged by the users.

**Directionality & Management**: Pairing has a *1-to-1 device relation*. Authentication of such a relationship could be either *unidirectional* or *bidirectional*. Unidirectional pairing achieves one device validating the other, not vice versa. This mainly depends on the implementation and the protocol. To achieve mutual authentication with unidirectional pairing, the user must repeat the authentication in the reverse direction. Conversely, bidirectional pairing achieves mutual authentication with one execution.

As the number of devices increases to more than two, the device relationship differs. Depends on the application needs, different network topologies can be adopted. A star topology network requires one device to be the proxy and every other device must associate with the proxy; in such a case, association is *centrally managed* by the proxy device and the device relation is *1-to-N* (where *N* is number of devices excluding the proxy). A network that employs a fully connected topology is a peer to peer network; management is *decentralised* (or *distributed*) and the device relation is *N-to-N*. In a fully connected ad-hoc network, every device needs to associate with every other device; thus, the number of association increases exponentially as the cardinality increases. Lastly, topologies like ring, line or tree require devices to associate only with their neighbouring devices; therefore, those topologies have the device relation of *1-to-n* (where *n* is the number of neighbours).

### *2.3.3 Location and Social Privacy*

In a house, different doors in different rooms for different purposes have different types of security locks. For example, a front door may have sophisticated locks that are difficult to break, while a door to the back yard may require less security as the backyard is protect by the fence, and some doors inside the house, like a kitchen door, may not even have a lock. The point is security requirements differ based on the location context. If an application is designed to be used in a risk free environment, authentication could be optional. For example, people save passwords on their personal computers (thus no authentication) because they know only trusted members can access the machines; conversely, people do not save passwords on public computers as the risk is high.

Here, we categorise three privacy settings: *Private* (or *Personal*), *Secluded Communal* and *Open Public*. A personal private area can be considered as a safe zone, a place where users trust and

know no intruders are within the space, and, within this space, minimum security is required. For example, if a user at home who wants to connect his mobile device to his home central system; the user should not need to go through a tedious authentication procedure. Instead, authentication should require minimum user effort. Next, we have environment that are confined by some physical features, like walls; however, not all confined spaces are private. Using an office space for example, the room could be confined by walls, yet it is not fully private as it is shared with other employees. We called such environment a secluded communal space. The level of threat inside a secluded communal space is minimal yet not risk free. A user may know who has access to this space but the user has no control over who is allowed to enter the space. Thus, the user has a certain level of awareness of knowing the surrounding environment but not able to control it. Finally, a user inside an open public space has no information about his/her surrounding peers. A cafe, for example, is an open public area. Random people can move in and out of the space; threat level increases, as devices in an open area are more vulnerable.

### *2.3.4 Distance*

Another variable that influences the design and usability of an association model is the distance of interactivity. The closer the distance between the associating devices, the more information about the devices would be available to the users. For example, it is simpler for a user to identify a device if the device is immediately in front of him/her than a distance away. We therefore identify three categories to define the distance of interaction: *Reachable, Noticeable* and *Remote* (or *Recallable*).

A reachable distance has the property of having the associating devices within reach of the users. Users can use their touch sensory system to interact with the devices. For example, a method that uses near field communication to associate devices [27] limits the interaction to be within a reachable distance. A noticeable distance has the attribute that the distance is not reachable, yet the associating devices are separated within a noticeable distance. For example, pointing a light beam and using audible sound are association methods that limit the interaction within a noticeable distance. For completeness, ideally, devices that are separated beyond a noticeable distance, such as occluded by a wall, could still be connected remotely. However, if the devices are separated by a remote distance, the device cannot associate in a spontaneous manner, as the user is required recalling prior knowledge of the remote target devices to identify them.

Furthermore, we should note that the distance category is related to the perceptibility category (section 2.2.3). Tangible associations require the connecting devices to be within a reachable distance so that the user can hold the devices. Also, sensible and non-interactive association models require devices to be within a noticeable distance to be associated.

## 3. CONCLUSIONS AND FUTURE WORK

Many new models for associating device have emerged, and each model was designed to solve a specific problem. Comparison between the models is difficult, as insufficient generalisation has been discussed in research. To understand differences and similarities between different models, we need classifications where we can allocate the models into categories. In this paper, we surveyed different association models from literatures and, based on those models, we constructed a classification of categories that influence the usability of the models. By using the classification, it is possible to identify and to build usability analysis of how association models differ or are related.

The focus of our paper is on usability; the work we introduced only represents a partial picture of generalising device association. Many more categories are yet to be identified. For example, *ease of learning* can influence users' willingness to adopt an association method. As well as, *adaptability*; some authentication methods like text input, gestures entry and biometrics are adaptable for both user and device authentication, while some are only suitable for one. People's prior knowledge and experience with an authentication model can influence their preferences.

For future work, we intend to adopt the classification in building a comparative usability analysis as well as taxonomies of different device association models proposed in literatures.

## 4. ACKNOWLEDGEMENT

This work is supported by the NoE Intermedia, funded by the European Commission (NoE 038419)

## 5. REFERENCES

[1] APPLE. Nike+iPod. http://www.apple.com/uk/ipod/nike/.

[2] BALFANZ, D., DURFEE, G., GRINTER, R. E., SMETTERS, D. K., AND STEWART, P. 2004. Network-in-a-box: how to set up a secure wireless network in under a minute. In *SSYM'04: Proceedings of the 13th conference on USENIX Security Symposium*, USENIX Association, pp. 207–222.

[3] BALFANZ, D., SMETTERS, D., STEWART, P., AND WONG, H. C. 2002. Talking to strangers: Authentication in ad-hoc wireless networks. In *NDSS'02: Proceedings of the 2002 Network and Distributed Systems Security Symposium*.

[4] BEIGL, M. Point & click - interaction in smart environments. 1999. In *HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, Springer-Verlag, pp. 311–313.

[5] BICHLER, D., STROMBERG, G., HUEMER, M., AND LÖW, M. 2007. Key generation based on acceleration data of shaking processes. In *UbiComp 2007: Ubiquitous Computing*, Springer-Verlag, pp. 304–317.

[6] BLUETOOTH SPECIAL INTEREST GROUP. 2006. Simple pairing whitepaper (revision v10r00).

[7] BUHAN, I., BOOM, B., DOUMEN, J., HARTEL, P. H., AND VELDHUIS, R. N. J. 2009. Secure pairing with biometrics. *International Journal of Security and Networks 4*, 1/2, 27–42.

[8] CASTELLUCCIA, C., AND MUTAF, P. 2005. Shake them up!: a movement-based pairing protocol for cpu-constrained devices. In *MobiSys '05: Proceedings of the 3rd international conference on Mobile systems, applications, and services*, ACM, pp. 51–64.

[9] FITTS, P. M. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology 47*, 6 (June 1954), 381–391.

[10] GEHRMANN, C., MITCHELL, C. J., AND NYBERG, K. 2004. Manual authentication for wireless devices. *RSA CryptoBytes 7*, 1, 29–37.

[11] GOODRICH, M. T., SIRIVIANOS, M., SOLIS, J., SORIENTE, C., TSUDIK, G., AND UZUN, E. 2009. Using audio in secure device pairing. *International Journal of Security and Networks 4*, 1/2, 57–68.

[12] GOODRICH, M. T., SIRIVIANOS, M., SOLIS, J., TSUDIK, G., AND UZUN, E. 2006. Loud and clear: Human-verifiable authentication based on audio. In *ICDCS '06: Proceedings of the 26th IEEE International Conference on Distributed Computing Systems*, IEEE Computer Society, p. 10.

[13] HINCKLEY, K. 2003. Synchronous gestures for multiple persons and computers. In *UIST '03: Proceedings of the 16th annual ACM symposium on User interface software and technology*, ACM, pp. 149–158.

[14] HOLMQUIST, L. E., MATTERN, F., SCHIELE, B., ALAHUHTA, P., BEIGL, M., AND GELLERSEN, H.-W. 2001. Smart-its friends: A technique for users to easily establish connections between smart artefacts. In *UbiComp '01: Proceedings of the 3rd international conference on Ubiquitous Computing*, Springer-Verlag, pp. 116–122.

[15] IWASAKI, Y., KAWAGUCHI, N., AND INAGAKI, Y. 2003. Touch-and-connect: A connection request framework for ad-hoc networks and the pervasive computing environment. In *PERCOM '03: Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*, IEEE Computer Society, pp. 20–29.

[16] KINDBERG, T., AND ZHANG, K. 2003. Secure spontaneous device association. In *UbiComp 2003: Ubiquitous Computing*, Springer-Verlag, pp. 124–131.

[17] KINDBERG, T., AND ZHANG, K. 2003. Validating and securing spontaneous associations between wireless devices. In *ISC 2003: Information Security*, Springer-Verlag, pp. 44–53.

[18] LAUR, S., AND NYBERG, K. 2006. Efficient mutual data authentication using manually authenticated strings. In *Cryptology and Network Security*, Springer-Verlag, pp. 90–107.

[19] LAUR, S., AND PASINI, S. 2009. User-aided data authentication. *International Journal of Security and Networks 4*, 1/2, 69–86.

[20] MAYRHOFER, R., AND GELLERSEN, H. 2009. Shake well before use: Intuitive and secure pairing of mobile devices. *IEEE Transactions on Mobile Computing 8*, 6, 792–806.

[21] MAYRHOFER, R., GELLERSEN, H., AND HAZAS, M. 2007. Security by spatial reference: Using relative positioning to authenticate devices for spontaneous interaction. In *UbiComp 2007: Ubiquitous Computing*, Springer-Verlag, pp. 199–216.

[22] MAYRHOFER, R., AND WELCH, M. 2007. A human-verifiable authentication protocol using visible laser light. In *ARES '07: Proceedings of the The Second International Conference on Availability, Reliability and Security*, IEEE Computer Society, pp. 1143–1148.

[23] MCCUNE, J. M., PERRIG, A., AND REITER, M. K. 2005. Seeing-is-believing: Using camera phones for human-verifiable authentication. In *SP '05: Proceedings of the 2005 IEEE Symposium on Security and Privacy*, IEEE Computer Society, pp. 110–124.

[24] PARK, D. G., KIM, J. K., SUNG, J. B., HWANG, J. H., HYUNG, C. H., AND KANG, S. W. 2006. Tap: Touch-and-play. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, ACM, pp. 677–680.

[25] PATEL, S. N., PIERCE, J. S., AND ABOWD, G. D. 2004. A gesture-based authentication scheme for untrusted public terminals. In *UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology*, ACM, pp. 157–160.

[26] REKIMOTO, J. 2004. Synctap: synchronous user operation for spontaneous network connection. *Personal and Ubiquitous Computing 8*, 2, 126–134.

[27] REKIMOTO, J., AYATSUKA, Y., KOHNO, M., AND OBA, H. 2003. Proximal interactions: A direct manipulation technique for wireless networking. In *INTERACT '03*, IOS Press, pp. 511–518.

[28] SAXENA, N., EKBERG, J.-E., KOSTIAINEN, K., AND ASOKAN, N. 2006. Secure device pairing based on a visual channel (short paper). In *SP '06: Proceedings of the 2006 IEEE Symposium on Security and Privacy*, IEEE Computer Society, pp. 306–313.

[29] SAXENA, N., UDDIN, M. B., AND VORIS, J. 2008. Universal device pairing using an auxiliary device. In *SOUPS '08: Proceedings of the 4th symposium on Usable privacy and security*, ACM, pp. 56–67.

[30] SORIENTE, C., TSUDIK, G., AND UZUN, E. 2008. Hapadep: Human-assisted pure audio device pairing. In *ISC '08: Proceedings of the 11th international conference on Information Security*, Springer-Verlag, pp. 385–400.

[31] SORIENTE, C., TSUDIK, G., AND UZUN, E. 2009. Secure pairing of interface constrained devices. *International Journal of Security and Networks 4*, 1/2, 17–26.

[32] STAJANO, F., AND ANDERSON, R. 1999. The resurrecting duckling: Security issues for ad-hoc wireless networks. In *Proceedings of the 7th International Workshop on Security Protocols*, Springer-Verlag, pp. 172–194.

[33] SUOMALAINEN, J., VALKONEN, J., AND ASOKAN, N. 2009. Standards for security associations in personal networks: a comparative analysis. *International Journal of Security and Networks 4*, 1/2, 87–100.

[34] SWINDELLS, C., INKPEN, K. M., DILL, J. C., AND TORY, M. 2002. That one there! pointing to establish device identity. In *UIST '02: Proceedings of the 15th annual ACM symposium on User interface software and technology*, ACM, pp. 151–160.

[35] VARSHAVSKY, A., SCANNELL, A., LAMARCA, A., AND DE LARA, E. 2007. Amigo: Proximity-based authentication of mobile devices. In *UbiComp 2007: Ubiquitous Computing*, Springer-Verlag, pp. 253–270.

[36] VAUDENAY, S. 2005. Secure communications over insecure channels based on short authenticated strings. In *Advances in Cryptology – CRYPTO 2005*, Springer-Verlag, pp. 309–326.