Autonomy, Heterogeneity, Trust, Security, and Privacy in Mobile P2P Environments

Jari Veijalainen
Department of Computer Science and Information Systems, University of Jyvaskyla
P.O.Box 35, FIN-40014 Univ. of Jyvaskyla, Finland
veijalai@cs.jyu.fi

Abstract

Mobile P2P environments are emerging as a result of rapid expansion of portable terminals that are able to establish direct wireless communication links among themselves. These kinds of terminals are under the control of persons and allowing interactions between their terminals is subject to trust between the individuals and their desire to preserve and exhibit various aspects of autonomy against each other. In this paper we relate the concepts of node autonomy, heterogeneity and trust with each other, defining also numeric measures for them. We also investigate what kind of interactions (such as transactions, file exchanges) are possible at certain heterogeneity, autonomy and trust level between the terminals. Finally, we relate privacy and security aspects in the environment to trust, autonomy and heterogeneity.

Keywords*: Mobile P2P, autonomy, heterogeneity, trust, security, privacy*

1. Introduction

The world is experiencing a rapid expansion of mobile networks. The current number of mobile subscribers has already exceeded two billions and the mobile telecom industry estimates that the number of subscribers would hit four billions within the next 10 years

There has been a lot of research done in mobile peer-to-peer systems and different aspects analyzed. In this paper we study aspects of node autonomy, its various aspects and degrees, and its relation to heterogeneity and trust in a Mobile P2P environment. Our assumption in this context is that the nodes are small portable devices (including personal area networks (PAN)) and that a user controls the behavior of his or her device. There can be client and server components, databases, etc. implemented on them, and the necessary communication components.

2. Node autonomy, heterogeneity, trust, security, and privacy

2.1. Mobile P2P environment and its model

We assume in this context that the part of the world we are interested in consists of physical entities that can be distinguished from each other, have a unique identity, and clear borders. The structure and properties of the entities determine what kinds of interactions are possible between them. The interesting entities are human beings, wireless portable terminals, as well as wireless access points in some cases. Human beings have the most sophisticated interaction capabilities among the entities we are interested in. They are of relevance here, because we assume here that everything happens in close physical proximity and people can evaluate each other often also face-to-face for trustworthiness.

Autonomy in this context is always related with some kind of interactions between entities that happen across the outer border of the entity. We assume here that the interactions between terminals are based on short-range wireless communication protocols, such as Bluetooth or Ultra Wideband (UWB) or WiFi. Thus, the nodes are assumed to be in a close physical proximity (say 10-100 meters) while communicating, but each of them can also leave at any time physically (walk away with the user) and thus loose the connection to other nodes in the group.

In addition, physical storage devices can be exchanged, such as memory cards or sticks.

The environment we are concentrating here is depicted in Figure 1. There are two examples of the environment. In the upper setting a few terminals are communicating with each other through a short-range wireless link, such as Bluetooth, and two laptops exchange a memory stick.

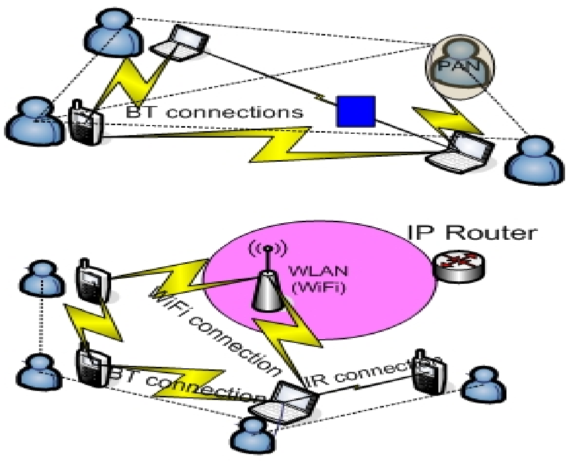

Figure 1. Two Mobile P2P environment instances; a) stand-alone (upper), b) with access to Internet (lower)

In the lower case there is also a WiFi access point and some devices use point-to-point infrared links. The dotted lines between users refer to direct interactions between human users, such as face-to-face conversations or mere watching on each other's behavior, or getting an eye-contact. In the upper setting a) we can assume that there are at most tens of people and all people can take up a personal contact, if necessary. In the lower setting b) it is assumed that the WLAN has access also to Internet that makes it possible to gather information from the past behavior of the other persons in the environment. In this case there can be hundreds of human users simultaneously in the environment and personal contact is not always feasible.

2.2. A model of the environment

Autonomy of an entity in the context of interactions means intuitively impossibility of external control on its behavior through interactions. In other words, if entity Ei can force Es to behave in the way it wants or prevent Es from behaving the way it does not want, then it can control Es' behavior through interactions. In this case Es is (fully) controllable – or not (at all) autonomous - towards Ei, and vice versa. A human can be killed by another human and thus the latter case takes effect, but under a threat of life often people choose the first option. Human-made machines, such as terminals, do usually neither recognize existential threats nor react to them, so they cannot be forced to behave in a certain way by threatening to destroy them. Destroying physically a terminal of course stops it to behave in any way, but the external world can also prohibit the terminal from taking certain actions without destroying it. A typical example is the prepared state (see section 4). On the other hand, it can force it to a certain state, if the structure allows it.

Closely following the incentive in [13]. Because we are mostly interested in interactions between entities, we model primarily *external and internal behavior* of the entities. External behavior towards other entities (the environment) is materialized as a sequence of actions. The entire structure of the entity is reduced to possible behaviors it can engage in. Because entities to be modeled are of finite size and complexity, they can only behave in finitely many different ways. In the entity model this means that each of them can only consist of finite number of different *behavior types* that in turn consist of sequences of *action types*. If we assume that each action instance has a non-zero duration and no behavior instance lasts infinitely long, then we can assume that each behavior type is of finite length, i.e. only contains a finite sequence of action type. We further note that different behavior types can have common prefixes. Thus, we model ith *entity type* through a finite, directed, labeled tree ETi, whose root node is empty and, each label on the arcs is an action type ATj, 0 < j < at_max+1, and the nodes contain the sequence of action types from root to the node itself, gathered from the action labels when traveled from the root to the node in the traversal order. Thus, each node contains the full action history, i.e. a *possible behavior* from the start (or birth) of the entity to the current *abstract state*. The leaf node contents, i.e. finite sequences of action types, are called *behavior types BTi1,..,BTi_imax*.

Notice that the modeling power of this model is that of a finite state automaton (FSA). If an FS automaton Ai can be represented as a directed acyclic graph (DAG) G(Ai), then for each path from the start state of G(Ai) to a particular end state of the automaton there is a corresponding path in ETi. It is formed by keeping the action types on the arcs of G(Ai) and inserting the action type sequences into the nodes, as above. If a node Nk has indegree Nk*in* larger than 1 in G(Ai), then such a node Nk is replaced by *in* nodes and on the outgoing arcs of each of these there will be all the action types that Nk has. In the node contents there is the sequence of action types that led to the state. Thus, the behavior type model remembers in its current state the path that was followed in the automaton. If there are loops in the graph representation of Ai, then each loop is unwound into a tree as above. Notice, that the outdegree of each node in ETi so constructed is the same as the outdegree of the corresponding node in the automaton G(Ai) and this is finite, because Ai is finite by definition. Of course, this process has no end and the length of the paths in the type tree will grow without bound, unless it is stopped at some point (see below). We can allow even infinitely long paths in the model, or paths of any finite length.

Some of the action types are *interaction types* that take part in interactions with the other entities, the rest are *internal action types*. An *inbound* interaction type event instances will wait for an *outbound* compatible event to trigger it, whereas an event instance of an *outbound* type does not need any inbound type of action to be enabled on an arc in order to be taken. This models the general structure of protocols and human behavior. Protocol messages can be sent without stopping to wait until the receiver has got the message, but rather, the sender moves on and begins to wait for the ack or other response. In a similar fashion a person can talk to another without being blocked, even if the other person does not listen.

Compatible action types and action instances are *liased* with each other in the triggering order. This is denoted by ATi@ATj. In ordering means that ATi must have happened before it can trigger ATj. In graph representation @ can be represented as an arrow from ATi to ATj that occur in different behavior types.

Projection of a behavior type BTik of entity Ei on interaction types is called (kth) interaction type, ITik. Two interaction types ITik and ITst are compatible, ITik@ITst, if there is a sequence of liaisons ATik1@ATst1,..,ATstr@ATikr such that each inbound action type gets triggered exactly once and there are no cycles in the graph formed from the BTIK,BTst and the liaisons. As defined, the model preserves *causality* i.e. a cause must precede effect. Notice that not all outbound action types need to be liased with any inbound type and that both entities can still reach the end of the behaviors BTik and BTst or continue the compatible behaviors infinitely.

Strong controllability (for BTik) of entity Es by Ei means that if Ei chooses to behave according to BTik, Es must follow certain behavior type BTst (ITik@ITst). *External controllability* (for BTik) means that if Ei chooses to follow BTik, Es can follow any behavior BTsu, where ITik@ITsu. *Strong autonomy* of Es against Ei (for BTik) means that if Ei chooses BTik, Es does not need to interact at all with Ei or can choose a behavior type BTsj incompatible with BTik. *Weak autonomy means* that Es can refuse interaction, but if it chooses to interact with Ei according to BTik, then it will be externally controllable by Ei (for BTik). Notice that two arbitrary entities Ei and Es do not need to have any compatible behavior types. A typical example are two terminals that do not support a compatible protocol stack (such as a Bluetooth stack).

The "life" of entity Ei is modeled as a tree LTi that contains all possible *behavior instances* entity Ei could go through during its existence. Root is empty and denotes the birth of Ei and the "life" advances by taking new and new actions that are enabled on the arcs. A finite behavior instance is a sequence of actions a1_iks a2_iks,.., ar_iks, where r is the length of BTik and aj_iks is of jth type in BTik. Subscript s denotes the sth instance of type BTik

The construction of LTi is similar to BTik, i.e. its root is empty, arcs are labeled with action instances ai_yth and each node contains the sequence of action instances from the root to the node. This sequence is called a real behavior, and denoted by Bi_h. Each real behavior Bi_h is a *shuffle* of the action instances in the order the corresponding action types occur in the corresponding behavior types.

. LTi could have infinite paths where finite behavior instances corresponding to one or several behavior types occur infinitely often or corresponding to infinitely long individual behavior types. But if we assume that each entity has a finite life time and that performing any action takes non-zero time, then there can only be a finite number of actions taken by the entity during its life time. Thus, each LTi can be considered to only contain paths of finite length, maximum being pl_imax. Further, because we assume that all entities we are modeling are finite, we can argue that the out-degree of any node in LTi is finite, say maximally outd_imax. Thus, there can be maximally only $pl\_imax^{outd\_imax}$ paths in LTi. The actual number of paths in LTi is denoted by #paths_i.

Although finite, the tree LTi can be huge in size. For instance, if there are maximally 10 parallel behavior instances allowed at any point in time (at any node), each of them can branch to two different paths from any node, and the life lasts 1000000 actions, the outdegree of each node would be maximally 20 and path length would be 1000000. Altogether the tree would have maximally 20^1000000 paths.

This modeling allows us to express potentially infinitely often recurring behaviors that are of the same type, e.g. Bluetooth communication sessions with other devices. It also allows to model allowed/prohibited parallelism of the behaviors, such as several simultaneous connections with different other devices and interactions with the user. This is expressed in the model by allowing the actions belonging to different behavior instances to be interleaved. Disallowing parallelism is modeled by the fact that none of the actions of instances of the same or different type are interleaved in any Bi_hx. Structural changes of the entity are expressed in this model by performing the actions belonging to "structure-changing type" and after the node where the structure change has been completed in the tree LTi behaviors of new type(s) begin to emerge. All behavior types must belong, though, to the behavior type tree.

2.3. Potential and effective autonomy

In general, we must distinguish between potential and effective autonomy. This is because there can be interaction types ITik between Ei and Es where Es is externally controllable against Ei (this set of interaction types is denoted by IT_ec(Ei,Es)), but also such types where this is not the case (IT_ea(Ei,Es)). Now, entity Es might be able to choose whether it appears controllable towards Ei or not, in a particular interaction of the latter type. Thus, although potentially Es would be able to behave against the will of Ei in any particular interaction of the latter type, it might want to cooperate and takes actions that Ei asks it to take, or refrains from actions Ei asks it not to take. Thus, Ei can follow ITik, but if Es so desires, it can choose another behavior ITsj and compel Ei to abandon ITik and follow ITij from some node in LTi. In this case ITik and ITij have a common prefix, as have also ITsj and ITsu.

Degree of potential autonomy of Es against Ei can be defined A_p(Ei,Es) = |IT_ea(Ei,Es)|/(|IT_ea(Ei,Es)|+|IT_ec(Ei,Es)|), where | | denotes the number of elements in a set. A_p(Ei,Es) = 1, if there are no compatible behaviors between Ei and Es

Let the paths in LTi be denoted by pi1,..,#paths_i. Degree of effective autonomy of Es against Ei on path pik up to the jth node $A\_e(Ei,Es)_{Pik|j}$ =

$|I\_ea(Ei,Es)_{Pik|}|/(|I\_ea(Ei,Es)_{Pik|j}|+|I\_ec(Ei,Es)_{Pik|j}|)$,

where $I\_ea(Ei,Es)_{Pik|j}$ denotes the number of interaction instances of one of the types in the set IT_ea(Ei,Es), where Es exhibited autonomous behavior and $I\_ec(Ei,Es)_{Pik|j}$ is the set of interaction instances where Es exhibited externally controllable behavior on path pik towards Ei in interaction instances initiated by it.

If $|I\_ea(Ei,Es)_{Pik|j}|+|I\_ec(Ei,Es)_{Pik|j}|) = 0$ then we let $A\_e(Ei,Es)_{Pik|j} = 1$. Further, it might be that at jth node there are incomplete interactions I_i@I_s between Ei and Es that belong to strong autonomy enabling types. If Es has exhibited autonomous behavior towards Ei, then these incomplete instances are counted into I_ea, otherwise to I_ec.

If jth node is the leaf node of LTi then we leave |j out of the notation. Notice that A_p(Ei,Es) and A_e(Ei,Es)$_{pkj}$ can be rather different for any path. This is because Ei can choose over time which types of interactions it starts and Es might be able to choose controllable or autonomous behaviors against Ei

Based on the above, we can define average effective autonomy between Ei and Es by

A_e(Ei,Es)av = . A_e(Ei,Es)$_{pik}$/|{pij1,..,pijk}|, where the sum is taken over all paths pij1,..,pijk, where Ei initiated at least one interaction attempt towards Es. A_e(Ei,Es)av = 0, if Ei did not try to interact during its life time with Es or if Es was controllable in all interactions and on each path there was at least one interaction between Ei and Es. A_e(Ei,Es)av > 0 otherwise and reaches 1 if Es exhibited autonomous behavior in all interactions with Ei.

Notice that A_p(Ei,Es) and A_p(Es,Ei) need not be in any particular relationship, unless we assume particular structures for Ei and Es, a special case being that they are identical. The same holds for A_e(Ei,Es) and A_e(Es,Ei) because of possibly different structure, but also because effective autonomy refers to the initiator of the interaction.

3. Node autonomy, trust, heterogeneity, security, and privacy

We define a node in this context to consist of a mobile terminal and a human user that uses it. Thus these separate entities form one entity, as concerns the perceptions of the environment of it. The potential interaction capabilities of a node towards another node consist of those of the human user (using any sense, natural language communication), as well as wireless data communication between terminals or exchange of memory sticks or cards. We assume that the user has a strong control on the behavior of the terminal, i.e. the terminal always does what the user tells it to do (if its structure makes it possible to receive and perform the command; e.g. command eat! Would not make its way to a terminal and even if it could “understand” what eating means, it could most probably not perform the action of “eating”). Further, full control means that the terminal does not perform actions that its legitimate user did not allow or order (cf. Trojan horses and other malicious software). The structure of the terminal can be fixed or modifiable only to a limited extent by the user, though (see below heterogeneity).

3.1. Node autonomy aspects

Node autonomy has four aspects in this context (cf. [13]).

The structure of the user and the device determine all possible conceivable behaviors they can exhibit towards each other and the environment. Whereas the structure of the human users is determined by the genes they got from their parents and the life history they have (including the mother tongue they learned, all emotional experiences they had,, education, experiences with various devices, etc.), the terminals are entities whose structure is determined by human beings while they are manufactured, deployed and used. Ideally, the human user in the node is the controlling entity and can freely decide on the structure of the terminal. Thus, *D(esign)-autonomy* allows the device hardware and software to be implemented, as the user wants, and he or she can store whatever programs and data he or she wants into its memory and in whatever format, irrespective of what other nodes (users) deem appropriate.

C(ommunication)-autonomy means that the device can start communicating and stop communicating with one or several other devices in the environment at any time using one or several protocol stacks. Here the autonomous behavior of node Ni against another node Ns is exhibited primarily by refusing to start communicating using a certain protocol stack (up to OSI level 4) used by Ni and stopping communicating at will. In terms of the model above, the action types are of form "Send a message of type MTi", "Receive a message of type {MTk, …,MTs}", "perform a suitable local action as a result of the received or sent message". Incompatibility of the protocol implementations where Ns would send unexpected protocol messages to Ni is deemed an error, not an indication of autonomy.

E(xecution)-autonomy means that a device can execute or refuse to execute an action request arriving from another device and can execute it as fast or slowly as it deems appropriate. Further, it can request other systems to execute some actions as a result of the original action request. This property is related with application layer protocol commands, such as FTP, or HTTP, 2PC, etc.

E- and C-autonomy can be subsumed by *M(anagement)-autonomy* that refers to the ability of the user to control the overall behaviour of the device, by configuring it to initiate, accept, or decline connections, store and send data upon (remote) user's request, shut the RF parts down, removing the battery, etc. M-autonomy also addresses the free movement of the user with the device. This causes it to leave one group of devices and perhaps to join another group to communicate within it wirelessly. The latter is called *roaming autonomy* of the node (R-autonomy). It gains importance in the environments at hand, because often people come to the space and leave the space and establish and close such short range connections during their presence.

3.2. Trust

and distrust are complex concepts and the definition depends on the environment (see e.g. [6] for further discussion). We maintain that trust and distrust guide the actual behaviors of an entity towards another entity. They are thus beliefs about the past and future behavior of the other entity. Whether an entity trusts or distrusts another entity is irrelevant as long as they do not have any intention or desire to interact neither directly nor indirectly with each other. Thus, only through real or intended interactions – or lack of them - trust gains its practical relevance; if Ei completely distrusts Es, then evidently it does not engage in interactions with Es.

Trust and distrust are often directly based on *reputation* of the other entity [1, 7, 10, 15]. In our environment the entities can be humans or terminals and the reputation is primarily attributed to the human being controlling the node.

Autonomy, as discussed above, characterizes behavior of an entity towards another from the point of view of an all-mighty observer that knows fully the structure of the entities. Trust, on the other hand, is a collection of beliefs entity Ei has about possible or actual behaviors of Es in the future or in the past, i.e. trust is a particular type of information a particular entity has about other entities.

An important point is that this information is very rarely complete; typically it is partial, especially in a mobile P2P environment under investigation here, whereas the entity structures are known to the human modeler to the necessary extent[1]. There are two reasons for this: only the external behavior of an entity can be observed by other entities, the internal behavior not. Second, only those external behaviors of Es can be directly observed that Ei can engage in with Es (due to their structure), the rest remain inexperienced for Ei. Information on the latter Ei can only obtain from other entities that had interactions with Ei – or that pretend that they have had them (cf. [7]).

Third point is that being partial and indirect, beliefs can also be false or true. That is, a belief of Ei that Es will behave according to behavior type BTsw towards Ei and BTij or other entities can be true or false. Only if Es is strongly controllable for BTij, then this belief will be true in all future interactions. If there are at least two different behaviors BTsw and BTsy, whose interaction behavior is compatible with BTij (ITij@ITsw, ITij@ITsy) then this belief does not need to be true. Ei cannot even determine the truth value in a finite life without further conditions on the structure of Es (see [13] for further discussion).

The fourth point is that trust and distrust entity Ei has towards Es might change over time, based on the interactions it has directly with Es and with other entities in the environment that mediate their beliefs about Es (cf. [6,DA06]). Whether entity Ei changes its view on trust and distrust on Es based on interactions with Ei or with other entities is also subject to node autonomy.

How to measure trust and distrust? There are many ways, but we suggest here the following. We enhance the nodes of the behavior type tree LTi by the beliefs Entity Ei has about the behaviors of entity Es. The set of beliefs in jth node on path Pik about Es is denoted by Bel(ik_j,Es). The individual beliefs are of form "In interactions with me (other entities) entity Es is trustworthy with probability ps (po) and untrustworthy with probability 1-ps (1-po)". Ei can change the probabilities in the beliefs in any direction as a result of interactions with other entities. An *irrational* entity can change the probabilities without any interaction with other entities, whereas a *rational* entity will change them only as a result of a direct interactions with Es or other entities that disseminate their beliefs about Es. A *stubborn* entity keeps those values that it assigns, when a belief is inserted into Bel(ik_j,Es) for some j.

The concept of trustworthiness might vary from entity to entity. This means that the same kind behavior might be deemed trustworthy or untrustworthy by different entities. Are there some general rules concerning this? What kind of behaviors would be considered trustworthy by all nodes? One common-sense criterium could be that the information an entity disseminates concerning its own behavior corresponds to its real external and internal behavior. Further, if an entity (node) disseminates information about the real or assumed behavior of other entities, it should not alter it from what it observed itself or obtained from other nodes. Finally a trustworthy entity should obey local laws and follow good manners, because criminals and impudent persons should not be considered trustworthy.

[1] Modeling of the entity structures needs only be performed to the extent it is necessary for the modeling task at hand. Only those aspects of the human behavior, for instance, are modeled that take place in the mobile P2P environment at hand.

Autonomous and trustworthy behavior of Es towards Ei is two different categorizations of the behaviors of Es by Ei. Evidently, (truly) controllable behavior is usually deemed trustworthy, because Ei knows that Es behaves as Ei tells it to behave and does not take actions that Ei does not want it to take. Autonomous behavior of Es towards Ei can be deemed trustworthy or untrustworthy. For instance refusing to start communicating with E can be deemed in both ways by Ei, depending on the context.

Trust and distrust are also related with autonomy in that sense that these beliefs regulate the degree of effectively autonomous behavior of a node towards other nodes. For instance, if the default assumption of node Es against any unknown node Ei is complete distrust, it would not even start wireless communication with Ei, i.e. it would exhibit effective communication autonomy in any particular communication instance. On the other hand, if the default assumption of Es against any Ei is complete trust, it would start communication at the first attempt and would evidently also perform the requested actions. The latter attitude of the Es can only be realistic if node Ei consists of a terminal controlled by a person trustworthy to Es.

3.3. Heterogeneity

Heterogeneity is a further dimension that is at the same time important when analysing the behaviour in the above environment. It refers to structural differences of the devices and to certain extent to differences between people. Heterogeneity at various system levels (device hardware, operating system, protocol stacks, data formats supported, data semantics, ontologies) is a consequence of design autonomy of the nodes in the sense that what is developed in isolation of other systems or entities tends to diverge with high probability and become structurally different and also non-interoperable. An example is UNIX operating system of which different research groups and companies developed diverse variants although they started from the same version released by Bell labs. Another example are natural languages and human cultures that greatly diverged during the human history.

After the above considerations, we have to point to the real power of an ordinary user to control the structure of his or her terminal(s). Because the terminals have complicated hardware and millions of lines of code in the operating system alone [9], the ICT skills of an ordinary user are far too weak to make profound changes possible. The only things an ordinary user can make are to load new pieces of software or update existing ones, perhaps install PCMCIA or memory cards and the necessary driver software –and download or input new data into the device. Thus, the most important way for a user to control the structure of the terminal is to select one among many different kind of terminals. Trust of the users that the officially announced behavior of the terminals corresponds to the real behavior is important in the selection. With this selection also the available short range wireless protocol stacks, such as WiFi, Infrared, Bluetooth, UWB or RFID become determined. Only into current laptops the user can add later wireless modems and their drivers that can change the communication capabilities of the terminal. These kinds of restrictions will be mitigated or removed, though, if the cognitive radio [11] hits the market.

Industry can diminish terminal heterogeneity, but currently it is limited to protocol stacks (BT,IrDA, WiFi) and XML-based interchange formats, as well as some multimedia formats (video, audio). On the other hand, the walled gardens erected by various operators make the devices largely non-interoperable with other devices, especially with those belonging to other gardens.

Measuring heterogeneity can be based on the above modeling incentive as well. If there are no common behavior types whatsoever between entity Ei and Es, then potential heterogeneity between Hi and Hs, H_p(Ei,Es) = 1. If Ei and Es are identical in structure, then H_p(Ei,Es) = 0. In general H_p(Ei,Es) is a number between 0 and 1 measuring the degree of difference in their structure, i.e. in their behavior types. Effective heterogeneity H_e(Ei,Es) = 1- (|{ITik : ITik@ITst for some t}|/|ITi|), where ITi is the set of all interaction types of Ei. As above, H_e(Ei,Es) = 1, if there are no compatible interactions and H_e(Ei, Es) = 0, if for any interaction type of Ei there is a compatible interaction type of Es. Notice that even if H_e(Ei,Es) = 0, H_e(Es,Ei) >0 might hold. This is because Es might have more interaction types than Ei. Also notice that from structural identity it does not follow that entities can interact in mobile P2P environment, because they need not have wireless (short range) communication capabilities. If they both have, though, one wireless short range protocol stack, then it is quite sure that they can communicate over it.

3.4. Security and privacy

Security refers to various issues. In this context the threats against the node are of two different type. Physical threat can materialize as an attack against the user him/herself, or against the terminal – or both. Wireless attacks happen over the short-range link to the terminal. A physical attack against the terminal can happen by taking it completely out of the control of the user, i.e. stealing it, or injecting malicious software into it – or opening a backdoor access - while having a transient control over it. Wireless attacks can happen through vulnerability in the terminal software or hardware that are innate or caused deliberately by an attacker or accidentally by the user.

Although terminal hardware also presents some value that warrants protection, much more valuable is information stored into them, be it data or programs. Information security means confidentiality, integrity, and availability. The above attacks can jeopardize any of them. Of course, availability is lost, if the terminal is stolen. How well confidentiality and integrity are protected depends mainly on the access control protection (password/PIN/fingerprint/voice) to the data and on the encryption applied to it. One can also conceive more radical protection measures, such as self-destruction of the terminal or its memories, if it or the user detects an attack.

Privacy can be seen as different from security, although they are closely related. In general, privacy addresses the question which other people are entitled to obtain what information about a particular individual and how can the individual control the information flows about him/herself. This is regulated in different ways in different countries and the sphere of individuality is larger or smaller, depending on the culture. In EU, for instance, releasing the location information of the person is under his/her control. Thus, e.g. tracking by another person or company must be explicitly authorized by the user. On the other hand, anybody in Finland can get to know through an SMS service the owner of a particular car, be it a private person, company, or government. The input to the service is the license plate information of the car in question that is of course publicly available. In Germany, only authorities can establish the connection between the owner and the license plate.

It is obvious that if the confidentiality of personal information is jeopardized, this can result in privacy breach through the attacks above. Further, if the integrity of the personal information is destroyed and the distorted information spread around, this can have nasty effects for the individual. A special case is digital identity theft or credit card theft. This could happen in the context of some top-level terminals that contain the credit card information, such as F904i of DoCoMo [8]. Also such simple things like whom the person has called can be very awkward, if published to a wide audience. Not to speak about the recorded phone calls that the current terminals also make possible.

What is the relationship between security, privacy, autonomy, heterogeneity and trust? First, node autonomy, or more precisely the autonomy of the user as a citizen of a country is more fundamental than security or privacy. The user has and must have the right to require security for the device and information stored into it and must have legal means to enforce security against other individuals or authorities. In other words, autonomy is based on individual rights incorporated into national legislation. The means to protect the individual sphere resulting from the individual rights include encryption and authentication subsystems, key exchange protocols, certificates, etc.. In technical terms security is most closely related with D-autonomy that addresses the capability of the user to determine the structure of the terminal. The structure must be such that information security can be guaranteed against all conceivable threats. Effective C-autonomy also can play a role, if the user protects information security by refusing totally to communicate with certain other nodes. In a similar fashion he or she can refuse to execute some wirelessly communicated requests ("send this file", "install this program") if there is a threat that this would lead to a security or privacy breach.

How about trust? Security and privacy are important aspects that the individual considers as he or she evaluates trustworthiness of another node. A primary privacy question is, "do I dare to mediate this private or secret information to this node?" And further, can I be sure that it is not forwarded in the original or in a modified form to somebody else?" It is worth of noticing that the interactions in a MP2P environment only can jeopardize privacy in part, because neither all private information is stored at the mobile terminal nor is it linked to it or its state. For instance, patient records are typically stored in databases of hospitals and cannot be accessed at the terminal. But stealing the terminal might open also these to the impostor, if the terminal grants access to the hospital database. Physical protection of the terminal is therefore important both for security and privacy. Other security aspects address the issue, whether the entire communication event and/or the transmitted data is kept secret from others than the intended recipients. In a wireless P2P environment it is rather difficult, although not impossible, to avoid communication event, such as file transfer, to take place so that it remains unnoticed by the other nodes. This is especially the case if radio path is used for transmission, whereas hiding the communication event might be easier, while using infrared. If a third party is able to listen to the traffic, then it should be encrypted. Bluetooth provides this, but there is a variety of opinions about the effectiveness [2].

4. Some analysis of the environment

4.1. Human to human interaction

The specialty of the environment at hand compared with e.g. E-commerce environments or P2P environments in global Internet is that the human have the possibility to have a direct face-to-face contact with each other. They can have in addition indirect reputation information that is gathered before they enter the meeting room or while they are there (especially in Fig. 1b) setting, from the Internet.

4.2. Borrowing the terminal

Current mobile terminals are not well protected against misuse in such a situation, where the legitimate user has rendered control to another person. This is because the physical access provides the possibility to go around security and authentication barriers, especially if the terminal is enabled by the user. Therefore, the action requiring perhaps the highest trust is to borrow the terminal to another person.

4.3. Communication behavior of the nodes

Users determine the communication behavior of the terminals based on their preferences and direct interactions with the other users. Thus, the nodes can select the used protocol stacks and the communication partners. Wireless communication and data exchange will only succeed if the terminals support interoperable protocol stacks up to the application layer, i.e. their effective heterogeneity must be less then one. Trust and distrust between entities (humans – reflected into the behavior of terminals against other terminals) guide to a large extent – but not entirely - their communication behavior.

A more complex case is when the nodes form a mobile ad-hoc network (cf. Fig. 1a). In this case a node can mediate traffic between two or several other nodes. This situation addresses autonomy and trust between a set of nodes.

Cognitive radio [11] might be important in the future in the environment while making the terminals generally interoperable.

4.4. File exchanges

A typical interaction types in the environment at hand are file exchanges. These can take place either through a wireless connection or by exchanging memory sticks or cards. For the previous case effective heterogeneity level is as above. For the latter case the terminals must use in the practice the same file system on the memory stick, otherwise it is not usable by both terminals. Thus, the potential heterogeneity between terminals can be larger in the former case.

As concerns trust, the node Ni receiving the file must believe that node Ns has right to hand over the file and must believe that the contents is not harmful, i.e. must have trust in the honest behavior of the other node. Digital rights management (DRM) is a system that attempts to increase the trust of content owners towards the nodes. It restricts possible behaviors of the nodes by disallowing sharing of contents not desired by the content owners or distributors. Currently the relevant contents are primarily music or videos. DRM implementation requires certain common (standard) structures at both terminals.

4.5. Database queries and transactions

Database queries expressed by a query language (such as SQL) are possible between nodes, if at least one supports a database (server) and a query interface that is accessible over short-range wireless communication facilities. If the trust level at the database server node Ns towards the querying node Ni is not high enough, executing the query might be denied altogether or results pruned. Thus, Ns would exhibit effective E-autonomy towards Ni. Allowing updates to the local database at Ns by Ni might require even higher trust level between Ns and Ni than allowing queries. An important ingredient in both cases is authentication of the node Ni by Ns. In file exchanges this is not absolutely necessary.

Traditional distributed transactions and 2PC protocol [4] could also be used among several nodes that support it. Effective heterogeneity between nodes is necessarily smaller than in the case of mere queries, because participating nodes must support both queries and the 2PC interface. As concerns trust, it must be higher than in the previous case, because 2PC protocol is a blocking protocol. This means that the coordinator node can block resources (data) accessed by a local sub-transaction at the participant nodes by refusing to send Commit or Abort message to the participants after a Prepare message. Further, if a participant node leaves the group (cf. R-autonomy) while in the Prepared state, the transaction cannot be terminated properly, because it might not be reachable in the future for the coordinator or other participants.

A similar problem is present also in transaction models that allow unilateral commit or abort for local sub-transactions [13,14]. In the previous case the backward recovery transaction (compensating transaction) cannot be started at the participating node, because the node might not be reachable when the time comes to start it (R-autonomy, C-autonomy). In the case of allowed unilateral abort [14] the necessary forward recovery is possible, but the fate of the global transaction remains unclear at participating nodes.

In general, if the duration of the transaction is longer than the feasibility of the communication links in the group, then these kind of problems concerning proper termination of transactions will occur. One can also ask, whether transactions make sense at all, unless the nodes participating in them are fully controlled by the coordinator. In this case they would not roam away during the transaction execution and would honestly perform the necessary actions prescribed by the used protocol (it is of course possible to lie about the real behavior, e.g. by promising to abort the sub-transaction, but still committing it, etc.).

5. Related work

The autonomy ideas and the first version of the entity model have been presented in [13]. The definition of the autonomy measures was different from what is found here. The backgrounds of the model are labeled transition systems, modal logic, but especially LOTOS [5]. In a similar fashion as in LOTOS if several actions are enabled in a state, exactly one of them is non-deterministically chosen. If all of them are inbound then the entity cannot behave in any way, until at least one outbound activity of another entity triggers one of the actions. Therefore, in real systems usually there is always an error action that is enabled if nothing else can be done, or timeout action that moves the control from waiting to a new waiting or some other state. Difference to LOTOS is that synchronization in LOTOS is synchronous, i.e. compatible actions happen at the same time forming one action.

Recently, somewhat similar incentives as in this paper have been presented in [16] in that respect that the entity interface behavior have been modeled and the activity has been seen as a game. Because autonomy is about external control that both parties might try to impose on each other, game-view is important. It was also recognized in

[13], but not pursued further formally. The difficulty lies in casting the concepts describing the will of an entity in a suitable logic that also covers "always", "eventually", etc.

In addition, trust should be modeled within the same framework. As hinted above, one needs to model beliefs about behaviors of other entities and formal characterization of what is "good" and what is "bad" behavior. Real behaviors and assumed behaviors could be the basis for this modeling, but the belief system of an entity should be represented in a suitable logic.

Research on reputation and trust models has flourished since E-commerce took off after ca. 1995. A comprehensive treatment is e.g. [12] and more recent work can be found e.g. in [1]. In this paper we have mainly applied the earlier results on trust and reputation. The novelty is to combine trust, autonomy, heterogeneity, and security into the same coherent model.

6. Conclusions

We have analysed in this paper node autonomy, trust, and heterogeneity and their relationships in a mobile P2P environment with each other and with security and privacy. We introduced a semiformal behavior-based model in order to describe more exactly what autonomy means and how is it related with heterogeneity and trust. It addresses behaviors of humans and terminals. We also define measures for the potential and effective autonomy and heterogeneity, as well as trust levels. Perhaps one of the main results of the paper is that autonomy, trust and heterogeneity can be related within the same model in a natural way. Furthermore, security and privacy are also closely related with trust, as important aspects when considering trustworthiness of an entity. All aspects are based on behaviors of the entities. Mutual heterogeneity level determines what kind of interactions is possible between two entities what not. Potential and effective autonomy characterizes the behavior of an entity towards another entity, mutual trust and distrust explain why the entities choose to behave in a certain way towards each other. The model makes it also possible to describe the changes in the trust and the resulting level of exhibited autonomy towards other entities over time.

The applied model is actually a transition system. They are used as models of various modal logics [3]. Much further work is needed to formalize the logic level, because one evidently needs several different kinds of modal logic (e.g. deontic, temporal/modal) to describe trust and autonomy at the same time. Empirical studies would also be necessary to investigate the real behavior of people in the environment that would make possible to determine the actual level of trust and distrust and the reasons for it, as well as the relationship of the trust level and actions allowed.

7. References

[1] K.Aberer, Z.Despotovic, W.Galuba and W. Kellerer, "The Complex Facets of Reputation and Trust", 9th Fuzzy Days, International Conference on Computational Intelligence Fuzzy Logic Neural Networks Evolutionary Algorithms, Dortmund, Germany, September 18 - 20, 2006.
[2] M. Bialoglowy, "Bluetooth Security Review, Part 1,25.4.2005." *Security Focus*. Avalable at http://www.securityfocus.com/infocus/1830.
[3] Blackburn, P., F.Wolter, J.v.Benthem (eds), *Handbook of Modal Logic (Studies in Logic and Practical Reasoning)*, North-Holland, Netherlands, 2006.

[4] Bernstein, P., V.Hadzilacos, N.Goodman, *Concurrency Control and Recovery in Database Systems.* Addison-Wesley, Mass., USA, 1987.
[5] T. Bolognesi and E. Brinksma. "Introduction to the ISO specification language LOTOS." *Computer Networks and ISDN Systems*, North-Holland, vol. 14, No. 1, 1988.
[6] E.Chang, P.Thomson, T.Dillon and F.Hussain, "The Fuzzy and Dynamic Nature of Trust", S. Katsikas, J. López, G. Pernul (Eds.)*: Proc. TrustBus 2005,* LNCS 3592, .Springer-Verlag Berlin-Heidelberg, Germany 2005, pp. 161 – 174.
[7] Z.Despotovic, K.Aberer, "P2P reputation management: Probabilistic estimation vs. social networks". *Computer Networks* 50 (2006), pp. 485–500
[8] DoCoMo F904i handset. See http://www.nttdocomo.co.jp/english/
[9] Evans,D.S., A.Hagiu, R.Schmalensee, *Invisible Engines; How Software Platforms Drive Innovation and Transform Industries*, The MIT Press, Cambridge, Mass. USA, 2006.
[10] C. Lin, V. Varadharajan, Y. Wang, Y. Mu, "On the Design of a New Trust Model for Mobile Agent Security". S. Katsikas, J. Lopez, and G. Pernul (Eds.): *Proc. of TrustBus 2004*, LNCS 3184, Springer-Verlag Berlin Heidelberg, Germany 2004. pp. 60–69.
[11] Mitola III, J., *Cognitive Radio Architecture*. John Wiley & Sons, USA, April 2006.
[12] Mui,.L, *Computational Models of Trust and Reputation: Agents, Evolutionary Games, and Social Networks.* Ph.D. thesis, Massachusetts Institute of Technology, NJ, USA. December 2002; accessed on Jan. 9, 2007 at http://groups.csail.mit.edu/medg/ftp/lmui/computational%20models%20of%20trust%20and%20reputation.pdf.
[13] Veijalainen, J., *Transaction Concepts in Autonomous Database Environments.* GMD Bericht Nr. 183, R. Oldenbourg Verlag, Germany, 1990.
[14] J.Veijalainen, A.Wolski, "Transaction-based Recovery". Chapter 11 in: A.Elmagarmid, M. Rusinkiewicz and A. Sheth (eds.), *Management of Heterogeneous and Autonomous Database Systems*, Morgan Kaufmann Publishers, San Francisco, CA, USA, October 1998, pp. 301-350
[15] L. Xiong and L. Liu. "Peertrust: supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Transactions on Knowledge and Data Engineering,* 16(7), 2004, pp.843–857.
[16] L. deLuca, T. Henzinger, "Interface Automata," *Proc. of the 9th International Symposium on Foundations of Software Engineering (FSE*), ACM Press, USA, 2001; pp. 109-120.

Authors

Jari Veijalainen

Jari Veijalainen is working as a full professor at the department since August 1996. On Nov. 17, 1999 he was nominated to a permanent full professor position by the University of Jyvaskyla. He was elected to become the head of the department starting August 2, 1999, a position held by him also during fall 1997 and fall 2006. During 2004-2005 he was heading the Information Technology Research Institute at University of Jyvaskyla. He is also a Docent in Computer Science at the University of Helsinki, Finland since May 1996. He holds a B.Sc. degree in Mathematics (1978) and M.Sc. degree in Computer Science (1983) from the University of Helsinki, Finland, and Dr.-Ing. degree in Computer Science (1989) from the Technical University of Berlin, Germany. His previous positions include permanent and visiting teaching and research positions at the University of Helsinki (Helsinki, Finland 1980-1985), Hahn-Meitner-Institut Berlin GmbH (Berlin, Germany 1985-1987), GMD-FOKUS (Berlin, Germany 1988), GMD-IPSI (Darmstadt, Germany 1992), VTT (Espoo, Finland, 1989-1997), Fraunhofer-Institut FIT (Sankt Augustin, Germany 2000-2001), Waseda University, GITI (Tokyo, Japan 2003), and University of Potsdam, Hasso-Plattner-Institut, Germany (2005-2006).
His research interests include heterogeneous transaction management, transaction models and mechanisms for computer supported co-operative work, workflow systems and electronic commerce systems, formal modeling, mobile computing, software engineering, and mobile multimedia data management.