IST-002057 **PalCom**

Palpable Computing:

A new perspective on Ambient Computing

Deliverable 37 (2.1.2)

Revised conceptual framework for palpable computing

Section I

Due date of deliverable: m 35

Actual submission date: m 38

Start date of project: 01.01.04

Duration: 4 years

University of Aarhus

Editor: Morten Kyng

Revision: 2

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	PU

Integrated Project

Information Society Technologies

Table of Contents

Section II of Deliverable 37, which contains papers from Palcom relating to palpability, may be found at:

http://www.ist-palcom.org/publications/deliverables/Deliverable-37-[2.1.2]-palpability-revised-SectionII.pdf

Section I: Introduction, vision and concepts

1 On the deliverable and its role in PalCom

This deliverable is the third related to the conceptual framework, a framework whose goal is:

- understanding and organizing knowledge of application domains for palpable computing in the form of a vocabulary and concepts to be used as a foundation for other activities.
- for users, it must provide means for understanding, learning, appropriating and using palpable computing systems.
- for designers and implementers it must support analysis, design, and implementation of palpable computing systems and be applicable through all phases of development.

The deliverable is described in the following way in the revised Description of Work (version 6.5)[1]:

> “This deliverable consists of a short overview consolidating the revised conceptual framework and a collection of papers (published and drafts) related to the vision and the challenges as well as the other terms and concepts used in characterising palpability. The main improvement over the previous version of the framework will be based on use.”

In the area covered by workpackages 2 to 6, i.e. the technologically oriented parts of the framework, a consolidated version of the framework has been produced. Due to the state of the PalCom toolbox and other software this consolidation is based on experiences from use by ‘PalCom people’. We are currently in the process of involving software developers from Milan, Italy and Tampere, Finland in the use of PalCom software and related concepts.

In the area covered by workpackages 7 to 13, i.e. the parts of the framework oriented towards use of palpable applications, a consolidated version of the framework has not yet been arrived at, but several proposals for revisions and additions are documented in the papers and drafts covered below.

The deliverable is the last before the final conceptual report. It is – as the predecessor D20 – meant to provoke and support the iterative, interdisciplinary and collaborative process of developing the notion of ‘palpability’.

In the period from now until month 48, where the final conceptual deliverable is due (D53), we will continue to use the document and develop the content – and we will focus on interaction with people outside PalCom in the ongoing development of the conceptual framework.

[1] The description of the deliverable has changed slightly over the last versions of the DoW before it was finalized in version 6.5. on 24 January 2007. Consequently the actual submission data for revision 2 (the current version) was not month 35 (November 2006) but month 38 (February 2007).

2 Contributors

The following people have participated directly in the production of the text for section I of the deliverable:

Büscher, Monika, Lancaster University
Kyng, Morten, University of Aarhus
Olsen, Jesper Wolff, University of Aarhus
Spring, Jesper H., L'ecole Polytechnique Federale de Lausanne

3 The Palcom Vision

Pervasive computing has taken computing beyond comprehensive systems into a multitude of devices and environments. In some sense this makes the computer disappear [10] and it enables ‘bricolage’ of disparate elements. However, people find it hard to realise the potential of pervasive computing. Which devices, services or resources are the best ones to use in a given situation? How to address breakdown? What to do when surrounded by potentially thousands of services and devices one could use? What when safety or privacy matters? To engage pervasive computing technologies effectively and creatively, people need to be able to notice and make sense of actual and potential computational processes, states, affordances and dependencies. They need to be able to do so in ways that are appropriate for their specific situation, their level of computer ‘literacy’ and interest. For us, an important element of what is needed is captured by the word ‘palpable’, especially in its meaning of 'plainly observable', 'noticeable, 'manifest, obvious, clear'.

To achieve palpability we originally complemented the vision of ambient and ubiquitous computing in six areas:

ambient computing	complemented with	**palpable computing**
invisibility		visibility
scalability		understandability
construction		de-construction
heterogeneity		coherence
change		stability
automation		user control and deference

Our work so far has shown that all these are indeed contributors supporting palpability. Especially ‘invisibility/visibility’ and ‘automation/user control’ has turned out to be pivotal with ‘construction/de-construction’, as captured in the ‘assembly’ concepts as the instrumental enabler. Other important concepts are the associated ‘service discovery’ and ‘inspectability’. Finally we have begun to look into the area private/public: we have found a strong need for making this area more palpable than it is today (see also [5]).

4 Towards palpable computing

People make palpability, when they are getting on with a job at hand, playing or experimenting with, when they enjoy, explore or bricolage together materials, environments, tools or technologies, and when they address failure or breakdown. Materials, environments, tools or technologies may be ready-to-hand, that is ‘invisible’ in the focussing on the job at hand, because they work well and we are used to them. At the other end of the spectrum, they may become present-at-hand, that is ‘visible’, demanding the attention of our senses, for example because they fail to support the work we seek to accomplish [4]. Thus palpability is not a property of materials, environments, tools or technologies in themselves. It arises and is negotiated in engagement with them. The ‘material’ used to build computing systems makes this process very difficult, and at the same time very important, because our ‘material’ is:

- *complex*
  To make computation work, complex, layered and abstract processes are required. In operation, non-experts struggle to perceive, let alone understand what is going on, or what affordances [3] might be available.
- *immaterial*
  Much of the computational processes and resources (e.g. networks) required for ambient computing are well outside people’s perceptual range, in ‘Hertzian space’ [1] and not easily translated or amplified.
- *New and changing*
  People have not had time to learn how to sense, perceive, understand and interact with ambient systems.
- *powerful but ‘dumb’*
  Computating systems can be immensely powerful, flexible, dynamic and interactive but they are inherently unaware. Systems can sense but not make sense.

Following [9], interaction with technologies can often usefully be understood as a form of 'material' conversation. From a human perspective, because people often have a flawed or limited awareness of system states/activities, they may find it difficult to know how to 'take turns' in this conversation and what to 'say'. 'Best guesses' may trigger system reactions, which may be interpreted, rightly or wrongly, as appropriate, absent, partial, unintelligible, or inappropriate system responses. These then inform the next human turn and so on, leading to potentially complex and hard to trace 'misunderstandings'. From systems' perspective, there is an equally or even more limited or flawed awareness of human states/activities on which to base the design of 'turns'. Furthermore, systems have no way of judging appropriateness of responses. If a workable collaboration can be achieved, people might make do, otherwise they may decide to re-design the system or discard it.

Palpable computing breaks into these cycles by making sophisticated causal relations, responsibilities, processes, failures, successes, services, data structures, affordances, (in)compatibilities, functions, emergent functions, activities, traces, possibilities, system actions, relations, dependencies, communications, changes, etc. available to the senses, addressing:

- *complexity* by allowing for inspection, interactivity, experimentation

- *immateriality* by manifesting/translating/amplifying both human and machine activities to make them perceivable
- *new-ness and change* by supporting learning and discovery, by anticipating, greater understanding and adeptness on both, the human and the system side
- *powerful-but-'dumb'-ness* by enhancing the range of machine perception used to inform behaviour and by manifesting/documenting machine reasoning

4.1 Design concepts

As argued in D20, how people achieve palpability can usefully be described with the help of the metaphors 'material', 'surface' and 'interaction'.

- *material*: Materials – the atoms and molecules – are the enabling conditions, the potential opportunities for making possible palpaple human action through tools and technologies. They shape the different affordances of environments, tools and technologies.
- *surface*: The senses take hold on the surface. It is here that material actions (changing friction, sounds, colours, smells, temperatures, etc.) communicate much about their dynamic internal processes and affordances. Surfaces are specific instantiations of the human action and of its effects; they are the spaces and times where palpability arises through atomic dialog.
- *interaction*: Interactions are the actual palpable dialog between people and technologies. People engage with other people, environments, tools and technologies with all their senses, moving in and out of different contexts. When interacting with the material world material actions are made sense of, reacted to, and are folded into, human embodied action.

This set of metaphors is still our current 'best guess' and does not seek to represent all-encompassing how the world 'really' is, but to formulate important features in a way that helps us describe and create palpable computing. In particular, the metaphors help us to understand, express and address the actively produced nature of palpability.

Figure 1 below presents a generic version of palpable computing delineating palpability in these terms. A specific example of a palpable system (the site-tracker) may be found in Deliverable 20.

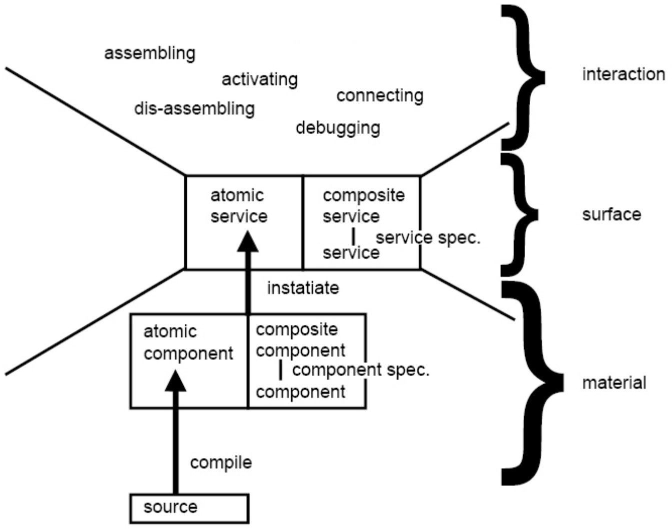

Figure 1. A schematic delineation of palpability in terms of interaction, surface, material.

Atomic and composite components – our design ‘materials’ are compiled from the source code and instantiated as atomic or composite service surfaces. Users interact with the services in ways that make more or less of the dynamic ‘material’ processes palpable.

Note that one way we support palpability is by giving materiality to ‘immaterial’ computation materials, e.g. through heartbeat, h-maps, query languages, USB lights, etc.

4.2 Human-technology relations

Through Palcom we move focus to the human-technology relations. In addition to Heidegger’s notion of ‘ready-to-hand’ that inspired Weisers formulation of Ubicom, we – and others – have argued for the application of the notions of ‘present-at-hand’ and ‘breakdown’.
If we also take into account more playful, aesthetic, constructive and creative modes of interaction (exploration and bricolage), it becomes clear that the focus should not be on the ‘result’ of interaction (e.g. ‘ready-to-hand-ness’) but on the means that allow people to achieve and negotiate such modes of interaction in their engagement with environments, materials, tools and technologies. In the future we intend to explore the mediation construct provided by the cultural-historical activity theory approach, as a potential means to gain a deeper understanding of how we can perceive/act through the mediation opportunities and their potential relations to the object-in-progress of our activity.

Empirical studies of the modes of interaction and PalCom design work continues to focus on:

modes of human technology relations	related terms	situation descriptions
ready-to-hand (Heidegger)	calm (Weiser) unremarkable (Tolmie) invisible (Ubicom) disappearing computer (DC) palpable (PalCom)	getting on with the job
exploratory-to-hand	palpable (PalCom)	play, experiment, explore
aesthetically-‘to-hand’	remarkable (Graves) palpable (PalCom)	enjoyment status (being seen with)
bricolage	palpable (PalCom)	dis-/assembling
present-at-hand (Heidegger)	remarkable (Tolmie) palpable (PalCom)	breakdown

Figure 2. Supporting palpability in human-technology relations

People generate different levels and forms of palpability in interaction with environments, materials, tools and technologies. One minute one might be happily getting on with one’s job (Figure 2), the next one may be distracted, begin to play with and explore the tools in use, discover previously unknown affordances (opportunities for interaction [3]), decide to change the assembly of environments, materials, tools and technologies, and do the job differently. The assembly might break down or fail to work as expected, at which point one would need to find out what is going wrong. One may also just enjoy (consciously or subconsciously) the beauty of environments, materials, tools or technologies. In short, people dynamically work and need to be able to work along a continuum of ‘visible/invisible’, depending on the situation of engagement.

Palpable computing gives technologies the means to respond to dynamic demands and it provides people with the means to dynamically negotiate situated use. This response/negotiation may involve (in)visibility, responsibilities, processes, failures, successes, services, data structures, affordances, (in)compatibilities, functions, emergent functions, activities, traces, possibilities, system actions, dependencies, communications, construction/de-construction, heterogeneity/coherence, change/stability, automation/user control, etc.

5 Concepts

In the following three subsections we briefly presents the PalCom concepts (in **bold**) and qualities (in *italics*) pertaining to materials, surfaces and interactions.

5.1 Materials: Concepts and qualities

Below key **concepts** and *qualities* pertaining to the material aspect of palpable computing are listed. These definitions are destilled from Deliverable 39 (2.2.2), where each is explained in further detail.

1st Order Resource
A low-level resource almost uniquely associated with a physical device, or node. Examples include processor load, memory, bandwidth and power.

2nd Order Resource
An abstraction used to describe those resources that either contain or consume 1st Order Resources. Examples include Assemblies, Services, Nodes, Components and Actors.

Actor An Actor is either a human user or some other system external to PalCom that can be perceived as a user of a PalCom Service or Assembly.

Assembly Descriptor
An Assembly Descriptor describes the organisation of a PalCom Assembly in terms of its constituent elements, what the assembly is intended to achieve, how it will achieve it, what preconditions are necessary and what output and/or postconditions will be affected.

Communication Channel
A Communication Channel provides a means of communication between Services. The concept incorporates the notions of communication medium (access type or bearer) and communication protocol.

Node A Node is a logical abstraction of PalCom Node and Non-PalCom Node that represents some item of computational hardware, i.e., a device.

PalCom Assembly
A PalCom Assembly is an organised collection of 2nd Order Resources composed in such a way as to deliver all or part of some application functionality.

PalCom Component
A PalCom Component is the basic unit of functionality, deployment and composition offered by the PalCom architecture. It is an encapsulated piece of code with well-defined interfaces describing preconditions and output.

PalCom Node

A PalCom Node is an item of computational hardware that hosts one or more PalCom Runtime Environments andwithin each, one or more components, services or assemblies.

PalCom Resource

A PalCom Resource is a logical abstraction of 1st Order Resource and Second Order Resoure and represents some element of a palpable system that is limited and consumable by other elements of the system.

PalCom Runtime Component

A PalCom Runtime Component is an executable instantiation of a PalCom Component.

PalCom Runtime Environment

The PalCom Runtime Environment provides the basic functionality required to host, execute and support the distributed communicative interaction of assemblies and the constituents thereof.

PalCom Service

A PalCom Service contains a PalCom Runtime Component coupled with the means to remotely communicate with other services, e.g., announcement, discovery, invocation. A PalCom Service is self-contained, can maintain state and is always expected to execute on a PalCom Runtime Environment.

Resource Descriptor

A Resource Descriptor is used to describe the characteristics of any entity that can be classified as a 2nd Order Resource (e.g., Node, Assembly, Service, etc.).

Service A Service is a logical abstraction of PalCom Service and non-PalCom Service that represents some item of remotely accessible, discoverable, and self-contained computational software.

Synthesized Service

A Synthesized Service is a service interface created by the composition of two or more Services as described by an Assembly Descriptor and contained in the corresponding PalCom Assembly.

Non-PalCom Node

A Non-PalCom Node is a Node that does not contain any PalCom Runtime Environments, but is otherwise capable of hosting Services.

Non-PalCom Service

Non-PalCom Service is a Service that does not contain any PalCom Runtime Components, but nonetheless offers an interface allowing it to be used within the context of a PalCom Assembly.

These key concepts are related as outlined in Figure 3.

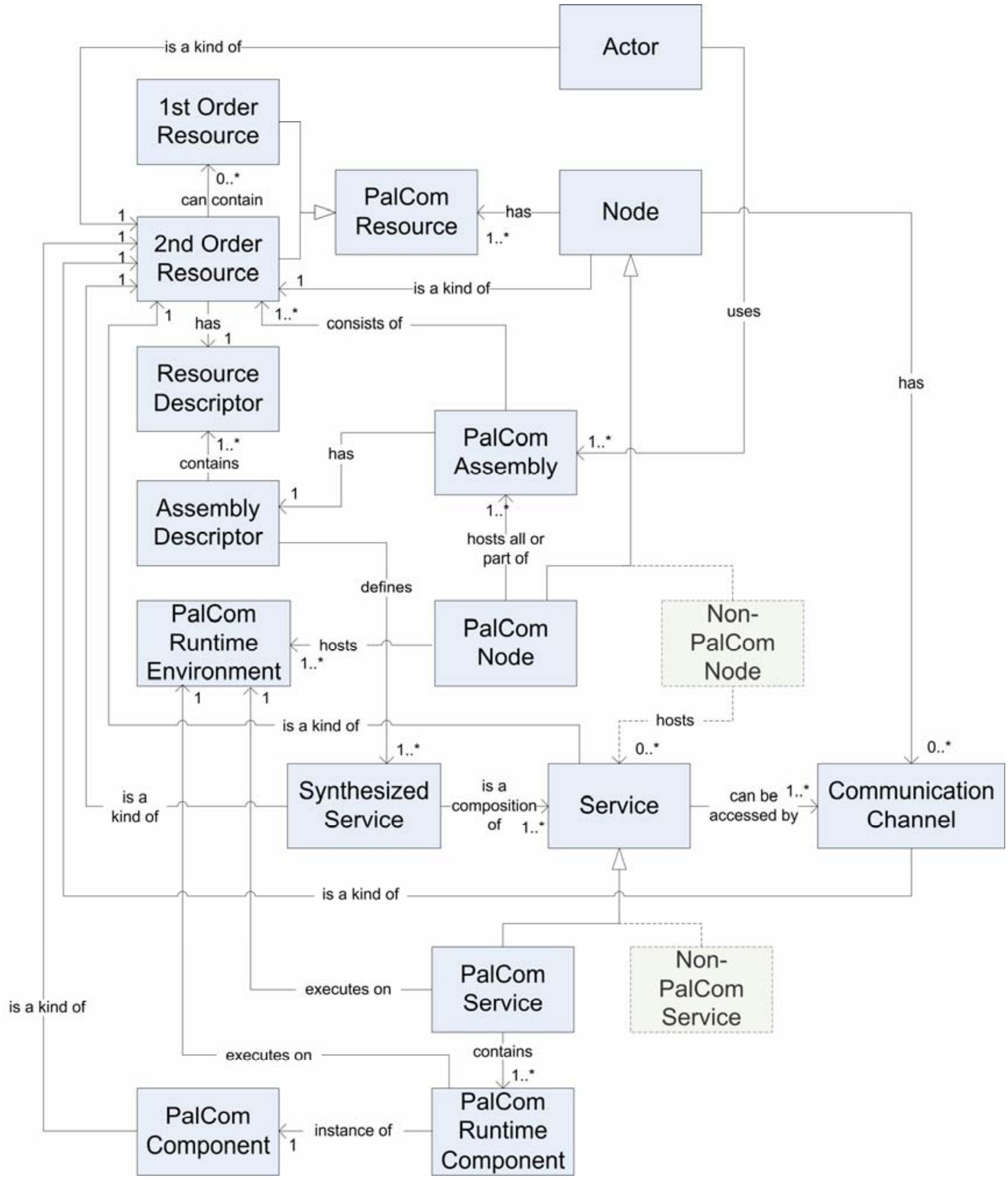

Figure 3. Concept map over key concepts. Boxes represent key concepts and lines denote relations

This section identifies the key qualities of the PalCom Architecture (materials) represent features recognised as cross-cutting concerns pertaining to materials that address the palpable challenges as stated in [7, chapter 4]:

Awareness The ability of a PalCom system to be aware of aspects of its own operation and usage context. The PalCom runtime environment is resource-aware by being able to accept, reject, or adapt the hosting and executing of software components. This resource-aware adaptation is based on reflection of the current resource requirements of the component as compared to knowledge of the resource constraints and configuration of the device on which it is running. A PalCom system or elements thereof (such as devices, assemblies, or services) can be considered as resource-aware if they adapt their execution according to the availability of resources at runtime. Similarly, an assembly, or elements thereof, can be considered as context aware if they are capable of reflecting on changes in the context of the computational environment and adapting their execution accordingly. This ability for being aware of these second-order resources plays vital role to the meet the challenge of Change and Stability. The PalCom architecture has the ability to track changes in the operating environment thereby enabling the PalCom system to adapt to new possibilities but also to detect failures. This again enables for the ability for the PalCom architecture to maintain stability in its operating environment by being aware of replacement functionality and enabling adoptation of this as needed.

Scalability The PalCom architecture is designed to be intrinsically scalable in order to cope with the complexity encountered in highly distributed, ubiquitous systems such as those enabled by the architecture. This scalability primarily translates into the ability to efficiently manage the distribution, availability and configuration of resources according to prevailing context and environmental conditions. This ensures that fluctuations in user/device populations and transaction volumes can be seamlessly managed without enduring performance degradation or interruption in service.

The ability of a PalCom architecture to scale up and down, however, puts enormous challenges on the way a user perceives the system – they way the user understands the system from what he/she sees. To encounter this, the PalCom architecture provides rich reflective possibilities to help the user to incrementally dive into the running system and locate the information needed with the right information level. Such incremental inspectability designed into the PalCom architecture supports the user in understanding the running PalCom system at any point in time.

Flexibility The PalCom architecture is designed to be flexible in terms of supporting runtime plugin, removal, configuration and reconfiguration of devices, services and assemblies. By allowing this, PalCom systems can adapt in form and function according to changes in operational context, environmental conditions and actor-specific behaviour.

This not only has the advantage of ensuring that applications can be dynamically (re)constituted, but also that components can be reused for different systems exhibiting common features. The mechanisms that control this flexibility are either user-triggered or triggered by automated system adaptation to events.

An important technique worth mentioning that enables such flexibility is loose-coupling, otherwise known as flexible-coupling in this context, that describes the manner in which palpable systems consisting of loosely interconnected dynamic populations of devices, services and actors allow more flexible operational modalities and easier ebb and flow of population members.

Finally, the PalCom architecture has been designed to enable flexibility by allowing a rich set of heterogeneous devices to collaborate in the formation of a PalCom system. This heterogeneity applies to both the software and hardware dimensions of the devices taking part. To support heterogeneity between devices, the PalCom architecture defines a set of platform-independent communication protocols acting as interfaces between the various soft- and hardware platforms. Any devices participating in a PalCom system must as a minimum be compliant to this set of protocols. Furthermore, different hardware platforms might provide very different resource capabilities (CPU power, communication abilities, display capabilities etc.) necessary for successful collaboration. The PalCom architecture handles such resource heterogeneity by abstracting away first-order (low-level) and second-order resources and representing them using standard entities, which are represented in a platform independent manner, and from this can be used for determining any constraints in composition of resources.

Usability Given that one of the critical aspects of palpabilty is the users' relationship to the computational environment, the PalCom architecture is designed to facilitate the means to render, manipulate and combine the functionality of user interfaces to achieve behaviour that fits and adapts to changing usage contexts and actor needs. An effort is made to ensure that the human actor comprehends and is in control of palpable systems with, and within which, he interacts. This includes exposing system operation transparently when necessary and opaquely when not, and accounting for automated system adaptation and configuration changes by clear and easily understandable means. The goal is to reach a point where the human actor feels that they comfortably and efficiently inhabit their palpable environment rather than act as an 'operator' as is more typical of conventional computational systems.

Resilience In addition to contemporary technical exception and error handling taking place locally and isolated within an application, the PalCom architecture supports distributed management of faults, failures and frailties by exposing causal traces and applying contingency procedures. This also applies to problems invoked by inappropriate and/or illegal behaviour by actors. The mechanisms supporting contingency management ensure that in the event of an erroneous or inappropriate condition, actions are available to adapt system behaviour to compensate in the most appropriate way. This might imply switching to an alternative network access point or service, using memory on another device, displaying information via alternative means, etc. A key issue is ensuring correct dependency tracing to guarantee that any and all affected elements of a PalCom system are either notified or automatically adapted to compensate for changes induced by error conditions. In compliance with the idea of palpable computing, such contingency management efforts may often involve the user by ensuring that she is supported in noticing and interrogating events and given the option to intercede in decision making that might otherwise be automated.

Security Security is a thread cutting across all aspects of a PalCom system. Although every entity associated with a PalCom system has specific security requirements, in each case these can be quite variable according to the particular usage context. Each entity will however make use of some combination and configuration of the following mechanisms:

- Certification: The ability to attach a certificate to an entity or message to aid in validation of its integrity, i.e. that it will do what it is expected to do.
- Encryption: The ability to encrypt stored and exchanged information.
- Authentication: The ability to authenticate an entity against established credentials. Previously unknown entities cannot be directly authenticated and must undergo another form of evaluation such as reputation analysis. This technique involves a trusted third party, known to both entities, that is willing provide guarantees based on previously observed and/or experienced behaviour patterns.
- Authorisation: The ability to apply permissions to actions that an entity can take, typically in the form of allow/deny statements. Permissions not only provide a layer of control over the actions an entity can take in situational contexts, they also act as the means to create a framework for structuring the relationships between the entities, including human actors, of PalCom systems.

Palpability is supported by allowing people to notice and inspect the security policies being applied. In addition we are looking into how the consequences of applying (and removing) security policies can be made inspectable (see also [5]).

5.2 Surfaces: Concepts and qualities

In this section we outline the key **concepts** and the materiality *quality* of the surface aspect of the PalCom architecture. Because action and material meet through the surface, the list of concepts contains two kinds of entries: entities and actions.

5.2.1 Entities

Environment An environment is, alongside the physical environment and all that this entails, a set of users, devices and services that a given user chooses to regard as a whole.

Device A device has a physical extent. It at least contains hardware with the ability to communicate with the PalCom environment. It furthermore has the capability to generate, hold or process information or a combination of these three. The user may or may not be able to interact with the device.

Service A service is a self-contained program, which is remotely accessible and discoverable by other services, developers and end users. It is instantiated and runs on a Device.

Assembly An assembly is manifested at runtime by a 'synthesized' service managing the composition of a set of other services, that is, dictating the required and optional services and the coordination between these. A PalCom Assembly can be either active by means of the synthesized service, or inactive by means of a stored XML description on some storage device.

State A state denotes the current condition of a given Service or Assembly, that is, is the service running properly, running faulty, not running at all and how is the current status of the Assembly, etc.

Connection A connection is denoting an association between two entities, i.e. a connection between assemblies, services and/or devices.

Route A route is denoting a set of connections arrangement as a string of pearls, where the pearls could be assemblies, services and devices. The string itself is one or more Connections.

5.2.2 Actions

Select The Select action is the act of choosing the devices and the services their offer, in the act of constructing an assembly that one wishes to build for a given situation.

Filter When one selects Services for a given situation, one may take advance of the action Filter. By filtering out different Services based on network access, location, type, availability or other characteristics to restrict the number of selectable services one may obtain a proper overview.

Connect The Connect action is the act of establishing a connection between assemblies, services and devices. This might be done either through automatically software programs or explicitly by the user themselves.

Assemble The Assemble action is the act of selecting, filtering, connecting and configuring coordination between assemblies, services and devices. The typical result of the action assemble is a newly created assembly, that is, either a passive description or a running synthesized service.

Inspect At any given time a user might want to take a closer look of the capabilities or state of a given Service, due the fact that he/she wishes to further investigate the functionalities of that services or to correct an erroneous situation. This is known as the Inspect action.

The key surface concepts are shown in Figure 4 below.

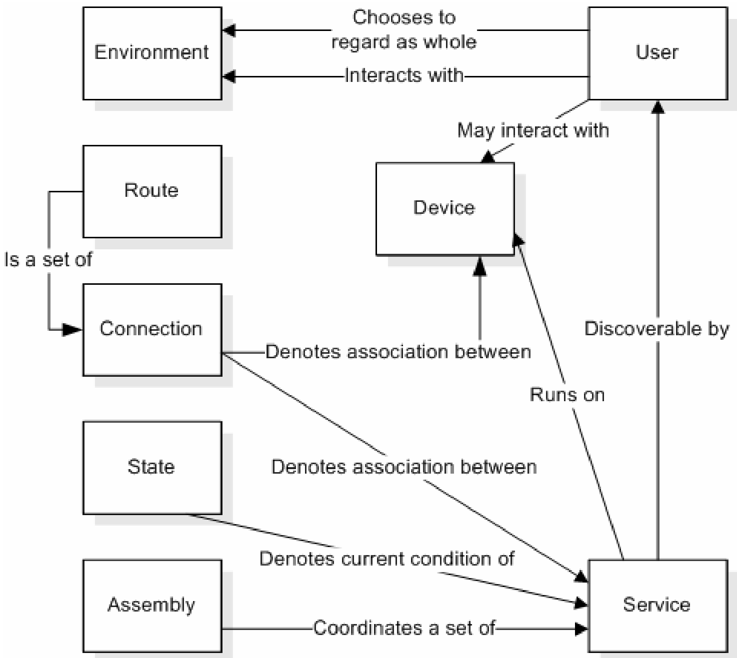

Figure 4. Concept map over key entities. Boxes represent key concepts and lines denote relations between key concepts.

The key quality of the surface aspect of the PalCom architecture is *materiality*. This is provided through a range of different inspection support tools or means. At runtime both Services and Assemblies possess an intrinsic ability to be rendered visible through the use of special browsers and/or debuggers. That is, they are able to manifest themselves in ways in which developers and end users are able to perceive; this supports comprehension. It is through the Surface aspect that the otherwise immaterial computational design material is given physical materiality. The concept of physical/digital materiality and how to make the digital palpable, as well as the concept of computational reflection is explored further in “Making the digital palpable”, “A Harmonization of Physicality and Digitality - Parallelisms at a Material Level” and “Towards Distributed Declarative Architectural Reflection”, see section II of this deliverable.

5.3 Interaction: Concepts and qualities

In this section we juxtapose **concepts** and *qualities* that people bring to human-computer interaction with those that traditional computing entities can muster. We define the palpable respecifications of these qualities that PalCom seeks to materialize in order to allow people to make computing palpable.

	Human	**Computing**
Context	People reflexively, indexically, intersubjectively and dynamically produce context, that is, as they fit into it, they simultaneously co-produce it. People sense context with all their senses and can produce appropriate action. Technologies are recruited into the reflexive and dynamic production of action-in-context.	Computing entities have an impoverished and relatively inflexible grasp of context. They lack the sensorial apparatus, cultural background and sentience to participate actively, they cannot sense what is going on (beyond the crudest parameters) or produce appropriate accounts of their states, let alone produce genuinely appropriate actions.

	Palpable Computing Palpable computing entities acknowledge the ultimate impossibility of machine context awareness and automatic appropriate action. They have a richer sense of context (e.g. through explicit interaction), they document their 'sensing' and 'reasoning about' context so that people can notice, understand, interrogate and manipulate it.	
Assemblies	People flexibly combine different technologies, materials, objects and artefacts to accomplish their tasks and activities. One of the most central challenges for PalCom is to design palpable computing devices/systems that can be manipulated and configured to work as resources in changing assemblies. [6, section 7.4]	Traditionally, computinge entities have been assembled into 'systems' by Designers. This demands blackboxing and complexity hiding.
	Palpable Computing Palpable computing seeks to allow people to notice and interrogate or break down the states, processes and affordances of all the computing entities around them. Rich documentation and tools support assembly, such as assembly browsers, but google-able libraries of previously tried and tested assemblies as well as multi-sensory interfaces.	
Traces and Trajectories	People are able to perceive and understand the traces of past actions and the trajectories for future actions mapped out by present moves – both in their interaction with social and material actors.	Traditional computing entities leave very few traces and provide only small scale feedforward.
	Palpable Computing Palpable computing seeks to support experimentation by allowing people to perceive traces of past actions and trajectories of future actions (for example the consequences of potential next moves).	
Documentary method of interpretation	People use a documentary method of interpretation [2], that is they take what other people, environments, materials, tools, technologies, etc. look like, sound like, wear, say, and do as reliable sign of underlying states, processes, intentions, emotions. They proceed to act on that basis, assuming that their human or non-human counterpart will use the same method.	Computing entities also use a documentary method of interpretation, but their sensory scope and interpretative ability is much smaller and less flexible than that of humans. This fact is often blurred or disguised in 'intelligent' technologies [9].
	Palpable Computing Palpable computing seeks to support the documentary method of interpretation by richly documenting machine actions and behaviours, by making more of human-documented aspects of context and human action available to the technology and by documenting its interpretations of such documentation..	
Reflexivity	Human action is reflexive in the sense that it is richly, retrospectively and prospectively, dynamically connected into a web of moves. Every move – a turn at talk, a glance, a pressing of a button, a keyboard operation – shapes,	Computing entities are incapable of reflexive action in a human sense – interpreting and re-interpreting their own and others' moves in the light of what has gone on before and is likely to occur next. Their actions are deterministic, although there may be very complex and very long

	and is interpreted in the light of every other move. People are aware of this (although not always consciously so), they can produce appropriate actions and actions designed to affect the production of situations.	cascades of causes and effects. The length, complexity and undocumented nature of deterministic cascades of causes and effects make it difficult for people to notice and make sense of significant moves.
	Palpable Computing	
	Palpable computing entities richly document actions, causes and consequences of actions, and chains of causes and effects. Documentation is tightly coupled to action, temporally, logically and geographically. How it is represented to users is specified through ethnographically informed, participatory design.	
Indexicality	Human actions are indexical, that is they receive meaning through their context. Indexicality or situatedness is not a problem for people but a powerful resource made available through the senses.	Because of the reasons described above, indexicality is only available in a very limited sense as a resource to computing entities.
	Palpable Computing	
	Palpable computing entities make it clear that they are incapable of making sense of action in context. They wear the fact that policies, limited context information, human or other machine actors' commands underpin each move they make.	
Intersubjectivity/ Interobjectivity	Most of what people perceive, feel, say, and do is intersubjective [8]. This means that under normal circumstances, people can assume that what they see, feel, say and do is understood by others in a similar enough way to allow interaction with them. We trust that most people are reasonable and truthful most of the time.	Most of the time people can assume that how manmade computing entities 'understand' the world is – in principle – discoverable and understandable. However, great complexity and variety are involved, making this very difficult.
	Palpable Computing	
	Palpable computing entities are designed to exhibit 'materiality' - 'standard' as simple as possible, tightly coupled documentations. This allows people to tap into scientific and everyday practices of making the natural and material world speak to them, using familiar, everyday practices of interrogating matter – such as comparison, multi-sensory experimentation, cause-effect experimentation.	
Orientation to rules	People actively orient towards rules, behaviour is not governed by them. The difference sounds subtle, but is momentous, because technology-behaviour is different. For people orienting to rules allows for improvisation in every move, without destroying the orderliness of social life and it means that socio-technical order can change through a myriad of moves that might comply, (mis)interpret, or break 'the rules'. Indeed, social rules are a byproduct of this ongoing production of order [2][9]. It is	Computing entities are rule driven. However the rules that govern their behaviour can be so nested and complex that it is difficult for people to determine which rules inform which behaviours.

	hard to make technologies that can fit into such dynamics.	
	Palpable Computing	
	Palpable computing seeks to make rules and rule following a richly documented, notice-able and inspectable aspect of machine action.	
Reciprocity of Perspective	The principle describes the fact that under normal physical conditions a person would know that if they put themself into another person's position, they would see the same as the other person sees. Reciprocity of perspective also applies to cultural 'environments', that is, to varying degrees, people are able to empathize with others points of view.	Computing entities cannot exchange physical or cultural perspectives with people. Their complexity and 'immateriality' make it difficult for people to 'put themselves in the shoes' of the material agencies involved or the designers' understandings and intentions inscribed in the material.
	Palpable Computing	
	Through rich documentary evidence and tightly coupled reflection palpable computing seeks to make it possible for people to sound out the perspectives taken by, and the scripts and scope for action inherent in, the material agencies of computing entities.	
Mobility	People move from context to context (e.g. home, work, hospital), constructing, deconstructing assemblies of technologies, objects, materials, and artefacts. This may involve physical movement, but could mean that the same set of people stays in the same place with the same set of technologies, but they change the form or focus of their activities. Many different systems and applications may be involved, and information may move in and out of digital and physical formats. [6, section 7.6]	Traditional computing entities are often embedded and assume that people need to move only a few artifacts (PDA, laptop). Configuration is often automated and triggered through context-information based policies. Interoperability is limited. This makes it difficult for people to address breakdown or creatively exploit the resources around them.
	Palpable Computing	
	Palpable computing supports mobility through contingency and resource management, by allowing people to RASCALize their devices and services if they wish to exploit automated adjustments, but at the same time wish to be in a position to make what is available to them, what is happening around them, and what they could do with the devices and services around them palpable.	

6 References

1. Dunne, A, and Raby, F. 2001 Design Noir: The Secret Life of Electronic Objects Berlin: August/Birkhauser
2. Garfinkel, H 1967 *Studies in Ethnomethodology* Cambridge: Polity
3. Gibson, JJ, 1986 *The ecological approach to visual perception*. Lawrence Erlbaum Associates, Hillsdale, NJ)
4. Heidegger, M. 1962 *Being and Time.* London: Blackwell.
5. Ingstrup M., Dourish, P. 'What are we interacting with?' In: Ingstrup, M. Towards distributed declarative architectural reflection. Department of Computer Science, University of Aarhus, Århus, September 2006; pp 81-102.
6. PalCom External Report 12: Deliverable 12 (10.1): *Initial Work Analysis Report*. Technical report, PalCom Project IST-002057, December 2004. http://www.ist-palcom.org/publications/review1/deliverables/Deliverable-12-[10.1]-Initial-Work-Analysis-Report.pdf
7. Palcom Working Note #66: Software Architecture: Contributions to Conceptual Framework Months 1-17 http://www.ist-palcom.org/cvs/cvs.php/palarch/workingnotes/wn66-sa-contrib-to-cf/wn66.pdf
8. Schutz, A. 1970 *On phenomenology and social relations*. Chicago: University of Chicago Press.
9. Suchman, L. (1987). Plans and situated action.
10. Mark Weiser, The Computer for the Twenty-First Century. *Scientific American*, pp. 94-10, September 1991