

# A cloud resource management model for the creation and orchestration of social communities

*Ahsan Ikram<sup>1</sup>, Ashiq Anjum<sup>2</sup>, Nik Bessis<sup>2</sup>*

*<sup>1</sup>University of West of England, United Kingdom*

*<sup>2</sup>School of Computing & Maths, University of Derby, United Kingdom*

*ahsan.ikram@live.uwe.ac.uk a.anjum@derby.ac.uk n.bessis@derby.ac.uk*

Abstract: Managing resources, context and data in mobile clouds is a challenging task. Specific aspects of spontaneity, large interaction space and dynamic interaction share a metaphorical resemblance to chemistry, chemical reactions and solutions. In this paper, it is argued that by adopting a nature-inspired chemical computing model, a mobile cloud resource management model can be evolved to serve as the basis for novel service modelling and social computing in mobile clouds. To support the argument, a chemistry inspired computation model, Chemistry for Context Awareness (C<sub>2</sub>A), is extended with Higher Order Chemical Language (HOCL) and High Level Petri-net Graph (HLPNG) formalisms. A scenario and simulation-based evaluation of the proposed model, focusing on two applications dynamic service composition and social communities identification, is also presented in this paper. The formal encoding of C<sub>2</sub>A validates its assumptions, enabling formal execution and analysis of context-based interactions that are derived using C<sub>2</sub>A principles.

## 1. Introduction

Mobile clouds and devices have seen extensive growth in capabilities over the past decade. The amalgamation of rich sensors, communication interfaces, processing capabilities, programming interfaces and a growing integration of sensors in physical environments, has generated a lot of ‘raw data/information/context’ in mobile clouds along with resource management and resource provisioning challenges. For these challenges, readers are pointed to [25] [28]. However, effective utilisation of this data, using existing context models will pose challenges and difficulties in aspects related to scalability, efficiency, usability and human computer interaction [16][17][18][19]. A growing number of users, devices and context sources renders computation in mobile clouds as primarily non-deterministic and dynamic, which makes existing deterministic and pre-defined computational logic and programming language models less applicable, thus restricting the scope of interaction. The increased need for scalability, extension, intelligence and adaptation, along with added dimensions of interactivity and social context, has pushed the mobile clouds resource management scope beyond the traditional context, such as human and nature inspired models.

Chemical computing has existed in the form of Gamma formalism since 1986 [1]. A Gamma program is a collection of reaction rules, made of an action and condition, acting upon a multi-set of elements until an inert state is reached (such that no more reactions in the solution are possible) [1]. From Gamma formalism and from chemistry-based inspiration, various other forms of chemical computing including chemical programming [2] have also evolved in the past two decades, including Chemical Abstract Machine (CHAM) [3], hmm-Calculus [1] and P-systems [1]. Dittrich et al. [4] presented a comprehensive review of chemical computing research and termed chemistry inspired computing research as ‘artificial chemistries’. Chemical computing models have also been used for non-deterministic and dynamic domains such as workflow and service orchestration [5].

An emerging aspect of personalisation is social awareness. Recently, considerable research has explored the realisation of social networks, computation and communities in mobile clouds (CCast [15], Cmobile [14], IperG [13], IPCity [12], CAMAR [11]). Mobile clouds provide a rich infrastructure for social computation and interactions as the sources, where the quantity and quality of context is richer in such environments. However, realising pervasive

1 social networks and interactions present contextual modelling, monitoring, visualisation and  
2 complex reasoning challenges for the research community.

3 Chemical reactions provide a natural metaphorical mapping to context-based computing. For  
4 instance, a context can be mapped to chemical elements, context interactions can map to  
5 chemical reactions, chemical solutions map to context compositions, and chemical properties  
6 map to the interaction capabilities of a context source. Therefore, by extending the similarities  
7 of structure, and then formalising these similarities and evolving a visual notation for  
8 representation similar to chemical notation, enables the realisation of a chemistry inspired  
9 context-based computing model, which is dynamic, scalable and spontaneous.

10 Formal representation of context and context-based interaction is critical for analysing the  
11 capabilities and standardisation of context models. Considerable research has investigated  
12 formal context representations, such as, Calculus for Context Awareness [21], Bigraphs [22]  
13 and XCML [20]. However, formalising a context model has been difficult, due to the  
14 variation in context classification and representation, together with large and growing space  
15 of possible interactions and dynamic context-based reactions. Nonetheless, a formal model  
16 strengthens the expressive capabilities of a contextual model, and opens it up for wider  
17 understanding and adoptability.

18 It is evident that information technology interactions in context-based mobile clouds have  
19 become complex. Chemistry inspired computation provides a metaphorical mapping of  
20 context, interactions, communities, visualisation and representation. It is proposed that if a  
21 chemical computing model for context-based mobile cloud can be evolved and formalised, it  
22 will serve as the basis for analysing the suitability and expressive capabilities of chemistry  
23 inspired computation for mobile clouds. This paper proposes a chemistry inspired  
24 computation model for context-based mobile clouds, and attempts to formalise and validate it  
25 using HOCL and Petri-net formalisms.

## 2. Chemistry for Context Awareness

26 In [1], [6], [8] and [7] we have presented a chemistry-inspired context modelling method,  
27 Chemistry for Context Awareness (C<sub>2</sub>A), and its components. In this section, we introduce  
28 the core concepts and components of the C<sub>2</sub>A model before applying a formal encoding to it.  
29 A Context Element (CE) is a basic unit of interaction and is a representation of characteristics  
30 of a context source. CEs belonging to an entity (a user or a device) are organised to form a  
31 Context Periodic Table (CPT). Every entity in the environment owns a CPT. The CPT is used  
32 to orchestrate and create a composite context, which is similar to compounds in chemistry.  
33 Context reactions (R), within CEs of a CPT, generate a smart space (Ss). Each Ss is modelled  
34 as a graph-based on the Context Periodic Table (CPT), where the CPT is associated with a  
35 user, group of users, or any entity in the environment (for example sensors). Furthermore, a  
36 Ss is either a Personal Smart Space (PSs) or a Social Smart Space (SSs), such that a PSs  
37 represents a single entity and a SSs represents multiple entities. To summarise, C<sub>2</sub>A is built  
38 upon the basic building blocks of Context Elements (CE) and Reactions (R). Context  
39 elements and reactions are contained and monitored by a set of algorithms that generate Smart  
40 spaces (Ss) [7].

### a. Context Element

41 A Context Element (CE) is the representation of a context source and is associated to an  
42 entity. A CE can be controlled and generated by the system and also by the user. For example,  
43 a system generated CE could be a GPS-CE (that represents the GPS coordinates of an entity  
44 using GPS sensor hardware) that is handled by the system and is based on a hardware sensor.  
45 An example of a user generated CE can be Status-CE (that takes the user activity as a textual  
46 input and stores it) that is created and updated by the user via an application or manually.

Table 1, an illustrative example of CE, presents a properties map of a ‘GPS-CE’. It gives an example overview with arbitrary values of how an active context properties table would look like at a given point in time.

GPS	
Physical Properties	Chemical Properties
Input = {(Address), (GPS Device)}	Status = {1}
Category = Location	Reliability = {1.0}
Title = GPS	Tags = {location, gps}
Type = {1}	Output = {X-Coordinate, Y-Coordinate}
Boolean = {true, false}	Composition = {}

Table 1 – An illustrative example of CE, a GPS Context Element

### b. Service Element

Service Elements (SE) are a special case of context elements, which represent services subscribed to by an entity. Building on the C<sub>2</sub>A principle, a new form of service representation is proposed in this research. A service is similar to a CE in structure, but with an added chemical property called ‘Signature’. Signatures are implemented in place of ‘Tags’, which is an optional property of a CE. A service signature is a tuple that has at most six sets within it, such that, each set stores zero or n values corresponding to each CPT column, see Table 2.

CPT Column	1	2	3	4	5	6
Service Specific Categories (Optional)	C1	C2	C3	C4	C5	C6
Value	Val <sub>1</sub>	Val <sub>2</sub>	Val <sub>3</sub>	Val <sub>4</sub>	Val <sub>5</sub>	Val <sub>6</sub>
<i>Example</i>						
Categories	location	<i>Empty</i>	Wifi	video	<i>Empty</i>	<i>Empty</i>
Value	berlin	<i>Empty</i>	BTOpen	CNN 23	<i>Empty</i>	<i>Empty</i>

Table 2 – Service Signature Structure

### c. Task Molecule

Task Molecules (TM) are similar to CE in structure. They are only differentiated in title because of separate placement in the CPT. Tasks are placed separately because the molecular structures consisting of more than one type of CEs are not considered in the generation of smart space graph, which is discussed later. Tasks are differentiated from services because of their scope; services in the C<sub>2</sub>A model are universal whereas tasks are local to an entity or user.

### d. Context Periodic Table

The Context Periodic Table (CPT) is a collective structured representation of Context Elements (CE). The CPT borrows its structure from Mendeleev’s periodic table [40] that serves as the basis of chemistry. In surveying context classifications it was observed that most of them classify context as specific context sources (for example location, time, profile), whereas this serves well for domain specific models. It faces scalability and extension issues if migrated or shared in a different environment. Alternatively, some projects such as CONON [32] and 5W1H [31] classified context as a higher abstraction, such as ‘where’,

‘when’, ‘who’ etc. However, they attempted the abstraction as an aspect of reasoning instead of context classification.

Similar conclusions were reached by Bazire and Brezillion [27], who in their extensive research of context classification proposed that “context can be specified for a given situation by answering the following questions: Who? What? Where? When? Why? and How? *Who* indicates, the agent of the action. *What* represents the object, the entity that sustains the action. *Where* and *When* give information about the spatio-temporal location of the considered action. *Why* gives the intentions, the goal (and eventually the emotions) of the subject. Finally, *How* makes explicit the procedure needed to realize the action” [27]. Table 3, presents a summary of context classifications observed (in various research projects and theses) during a literature review and where possible, mapped to the classification discussed above. If a classification is found that doesn’t map to any of the Where, When, How, What, Why, Who category it is placed in the last column. Notably, when various context classifications were mapped, it became evident that most of them map to one of the types of the 5W1H, 7Ws or 6Ws model [24], [26], [27].

	Where	When	How	What	Why	Who	Unmapped Context Classifications
Ambisense [28]		X		X		X	Social context
Brown et al. [38]		X				X	Season, Temperature
Dey & Abowd [34]	X	X		X		X	
Jang & Woo [31]	X	X	X	X	X	X	
Kaltz et al. [33]	X	X	X	X		X	
Kjaer [30]	X						Proximity
Pascoe et al. [36]	X	X				X	Conceptual state
Ryan et al. [39]	X	X	X			X	
Saimotion [42]	X	X				X	Content, Organizational
Schimdt et al. [37]			X	X			Applications, Social context
Zimmermann [35]	X	X		X		X	Relations

Table 3 – Existing context classifications mapped against 5W1H classification

Building upon these findings, we proposed a context management data structure called the Context Periodic Table (CPT) [7]. The CPT has six fixed columns irrespective of number, amount or type of CEs available. The columns are titled, Where(1), When(2), How(3), What(4), What(5), Who(6). The six columns are finalised based on the earlier context

classification research, such that, ‘why’ aspect is left out, considering the dynamic nature of mobile clouds. *Why* is expected to evolve or be deduced, and is not modelled as a source of context. Moreover, an additional aspect, ‘what’ is introduced to model context sources such as content, tasks and services, as this aspect was identified as absent in Table 3. These six columns map to the six numeric values in a CE structure under the property ‘type’ [7].

While introducing the CE structure, the property ‘category’ is used to add a specialisation to the CPT. Vertical columns can be further sub-divided into categories. For example, two columns under the ‘WHERE’ column of a CPT can have two categories that may represent ‘relative where’ and ‘absolute where’. Alternatively, there can be three categories representing ‘location’, ‘orientation’ and ‘direction’. A column is bound to the six basic columns under any circumstance, whereas the category-based specialisation is optional and varies from entity to entity. For instance, a device loaded with location and direction sensors might have two columns, whereas a standalone temperature sensor might only have one hardcoded location element under the ‘where’ column. The CPT structure discussed above can be summarised as follows:

$$\begin{aligned}
 &CPT = CE_1, CE_2, CE_3, \dots, CE_n \\
 &\text{where } CE_i = \text{A single context element} \\
 &\text{Type, } T = 1, 2, 3, 4, 5, 6 \\
 &\text{Category, } C = \{ C_{11}, C_{22}, C_{35}, \dots, C_{n4} \} \\
 &\text{where } C_{ij} \in T, j \ 1 \leq j \leq 6 \\
 &\text{FunctionMap: } CE \rightarrow T, \text{ where } CE \in CPT
 \end{aligned}$$

Listing 1 – Periodic Table Structure

#### e. Context Reactions/Bonds

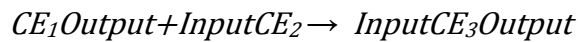
The CE and the CPT form the structural basis for the C<sub>2</sub>A model, and its backbone are the reactions that are defined by this base. Reactions (R) are the second part of the artificial chemistry and they define the circumstances, types and requirements that enable the interaction of CE. The CE and CPT structures, along with the requirements for pervasive interaction, serve as the basis for identifying the type and nature of bonds that could be realised using C<sub>2</sub>A. A bond represents a reaction, and it can be an expression consisting of integer, Boolean or strings (for example, true, false, =, <, >, ‘string’), a type of bond from a primary set of bonds, or a bond from an extended set of bonds. A primary set of bonds consists of one reaction for each column of a CPT and is defined as a *primary bond*. Primary bonds use chemical and physical properties of CEs to occur and execute. For example, using the input and output characteristics of the CEs it can be said that if the output of a CE matches the input of a CE, semantically and syntactically they could interact to form a bond (and call it *Interface Bond*), that shows that the output of one element could flow into the input of another to trigger a reaction. Table 4 presents a summary of reaction types before discussing them in detail.

Enabler	Bond Title	Description
CPT Owner	Association	Creates a bond between the owner of CPT and any other CE
Relevance	Abstraction	Creates bond between CEs of same type but varying relevance
Data flow	Interface	Bonds between CEs that have potential for input/output mapping
Mobility	Proximity	Represents the accessibility between two CEs based on

		proximity
Time	Temporal	To show the sequence and timing of an interaction
Services	Subscription	A bond showing the subscription when users subscribe to a service

Table 4 – A summary of possible context Reactions (R)

The above list of *primary bonds* is inspired from the operators evolved in XCML[20]. It is proposed that in the chemical reaction model, new types of bonds/reactions can be added to the model dynamically, called ‘secondary bonds’, and therefore, different solutions may have different sets of secondary bonds whereas they must have the same set of primary bonds. This is a novel aspect of chemical computing that renders scalability, extension and behaviour adaptation of a solution at runtime. In the following section a notation is introduced which is used to formally represent the reactions. Based on chemical reaction method of representing a reaction using LHS and RHS a simple interface bond can be summarised by a periodic table structure (cf. Listing 2)



where  $CE_3$  is a compound that represents  $CE_1, CE_2$

bonded by an interface bond

Listing 2 – Syntax of a reaction representation

#### f. Graphical Representation of Solutions

Chemistry is characterised by a graphical representation of elements, compounds and solutions. An important aspect of using a chemical inspired approach is the creation of a graphical representation framework. The reason for this objective is the hypothesis that a visual representation of context reactions and interactions will not only make the underlying context model expressive, but it will also enable a user’s interaction with context and its reactions, therefore, enabling the involvement of users in the process of context awareness, as opposed to users only being the consumers of context-based services created by developers. Visual representation also enables high-level abstractions of information and context. For visualisation, C<sub>2</sub>A adopts a basic graph-based structure where a basic reaction is visually represented as shown in Figure 1.

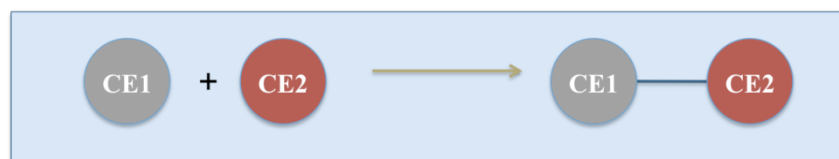


Figure 1 – Visual representation of a basic reaction

A CE is visually differentiated by the colour of its background, where the background colour is associated with a column *type* inherited from CPT. The colour combination can be altered as long as each column is represented with a different colour. The output of a reaction shows a connection between two CEs, such that, the texture, colour and label of the connection (edge) identifies what type of bond exists between two CEs. For example, interface bonds are represented with an arrowhead pointing to the direction of data flow.

Visual model is used to generate visualisation of a higher abstraction of C<sub>2</sub>A called ‘Smart Space’ (Ss). Ss are the proactive data structures of a C<sub>2</sub>A structure and by presenting smart space structure we cover both the representation methodology and the abstractions. A smart space is a graphical meta-representation that models both CEs and any possible reactions

1 between them. The smart space reaction mode is the autonomous reaction mode, and a novel  
2 aspect of the C<sub>2</sub>A model is its applicability and usability in end-user and developer mashup  
3 capability.

4 Smart spaces are of two types: personal (showing visual representation of context from a  
5 single entity's point of view) and social (showing visual representation from the perspective  
6 of a group of users).  
7

8 Personal Smart Space [7]: A Personal Smart Space (PSs) is a graphical abstraction of CPT  
9 (see Listing 3).  
10

$$11 \quad PSs = Graph(CPT)$$
$$12 \quad Graph(CPT) = (CEs, Bonds)$$

13  
14 Listing 3 – Personal Smart Space representation  
15

16 Social Smart Space [7]: Social Smart Spaces (SSs) is a collection of one or more PSs (see  
17 Listing 4).  
18

$$19 \quad SSs = Graph(PSs)$$
$$20 \quad Graph(PSs) = (PSs, Bonds)$$

21  
22 Listing 4 – Social Smart Space representation  
23  
24  
25

### 26 3. Formal C<sub>2</sub>A Encoding

27 This section presents a formal encoding of the C<sub>2</sub>A model using  $\gamma$ -based HOCL and Petri-  
28 nets. The process can be divided into two phases. In the first phase, HOCL is used to encode  
29 structure, grammar and syntax of C<sub>2</sub>A for writing reactions. The resulting structure is a form  
30 of sorted algebra that is integrated in the HLPNG grammar. In the second phase, the resulting  
31 petri-net model is used to encode graph-based smart spaces. HOCL was adopted for encoding  
32 C<sub>2</sub>A because of the common assumption of multisets and elements as enablers of chemical  
33 computing. Similarly HLPNG were used firstly, because they are built on a specification of a  
34 many-sorted algebra that is derived from the HOCL encoding, and secondly, because of the  
35 natural mapping of petri-net *Places* as *CEs* and *Transitions* as *Bonds*. We then test the  
36 evolved encoding using a storyline-based scenario.  
37  
38

39 Higher Order Chemical Language (HOCL) is a gamma ( $\gamma$ ) based calculi that is inspired by the  
40 chemical metaphor [10]. Gamma is a programming model where computation is seen as  
41 chemical reactions between data represented as molecules suspended in a chemical solution  
42 [9]. Gamma is perceived as the first chemical model of computation [1]. As discussed in [1],  $\gamma$   
43 originated as a multiset rewrite computation model, but it has seen various extensions in  
44 terms of its syntax and expression capabilities.  $\gamma_0$  was the original version of the language;  $\gamma_c$   
45 was the extension that enabled inclusion of conditions to the  $\gamma$ -abstractions and  $\gamma_n$  was the  
46 extension that enabled 'atomic capture', enabling  $\gamma$ -abstractions to extract multiple  
47 expressions from a solution. They were combined together in a higher order  $\gamma$  called  $\gamma_{cn}$ , in  
48 which HOCL is encoded. HOCL provides a higher order gamma abstraction for modelling  
49 computing entities to rewrite multisets, where the rewriting rules serve as chemical reactions.  
50  
51  
52

53 Petri-nets or specifically "High-Level Petri-Nets Graph (HLPNG)" [23] are used for  
54 modelling concurrency behaviour in networked systems. However, due to the graphical  
55 nature and features such as sequencing, merging and synchronising discrete events it can be  
56 used for distributed systems. Like bigraphs, HLPNGs have both graphical and logical forms.  
57 However, HLPNGs support many-sorted algebra, along with supporting terms and functions,  
58 which is beneficial in situation modelling and classifying context. Moreover the many-sorted  
59 algebra serves as the bridge between HOCL and HLPNG. The graphical notation consists of  
60  
61  
62  
63  
64  
65

four basic components: places, transitions, inputs and outputs. Both places and transitions have inputs and outputs. Places have tokens that are the modes of transition and activity. Transitions can execute logical functions on the tokens to check for success or failure of transition.

## C<sub>2</sub>A Encoding

Starting from a basic reaction, in this section we build on C<sub>2</sub>A concepts of reactions, elements and solutions by deriving their HOCL encodings, and later use the HOCL encoding to generate a petri-net model. The basic structure and grammar of HOCL is defined in [1] and [9]. The primary unit of interaction in HOCL is a reaction, represented as a  $\gamma$ -abstraction.

$$\gamma(P)C.M \quad \text{Eq. 1}$$

Eq. 1 shows an atomic chemical reaction encoded in HOCL as a gamma abstraction. It is read as ‘replace pattern P by solution M if condition C holds true’. This serves as the basic building block of HOCL and is the enabler that realises multiset rewriting. A solution in HOCL is represented with angle brackets, and commas separating constituents of a solution, therefore:

$$\langle 5,6,2,4,3,5,5,2,\gamma(x,y)x \geq y.x \rangle \quad \text{Eq. 2}$$

$$\equiv \langle 5^3,6^{2^2},4,3,\gamma(x,y)x \geq y.x \rangle \quad \text{Eq. 3}$$

$$\Rightarrow \langle 6,\gamma(x,y)x \geq y.x \rangle \quad \text{Eq. 4}$$

Eq. 2 represents a solution of natural numbers along with a single recurring chemical reaction, which compares any two numbers from the solution and replaces them with the larger of the two. This single reaction goes on recursively to the point that the solution becomes inert and there is no more reaction possible as shown in Eq. 4. Eq. 3 presents the same solution as that of Eq. 2, however, similar elements of the multiset are represented by a ‘multiplicity’ superscript. Multiplicity can be anything from negative numbers to infinity. An element with infinite multiplicity always remains present in the solution.

**Finite-but-large and growing space:** In previous sections we have highlighted that a formal model is desirable for context-based systems; mainly because of the finite-but-large set of behavioural options such applications and systems exhibit. This is handled in C<sub>2</sub>A by separating data from interactions and is encoded in HOCL as shown in Eq. 2, such that, the gamma-reduction rule that executes a reaction in the solution can take as input any two elements of the solution. For example, two independent sequences of reactions can be ((6,5), (6,2), (6,4), (6,3), (6,5), (6,2) -> 6) and ((5,5), (5,5), (5,5), (2,5), (5,2), (4,5), (5,6) -> 6). In the previous example, though the two example executions took different paths, they reached the same result. It is not necessary especially in the instances where the reactions are not n-shot (that are always present in the solution), however this may be possible in one-shot (the reactions that disappear after an execution). For example, if we place another reaction such that it replaces x with y and make it one-shot then the output could differ depending on when the one-shot rule fired and upon which dataset. If, in the execution pattern ((6,5), (6,2), (6,4), (6,3), (6,5), (6,2) -> 6) a one shot rule fires at the beginning on (6,5), 6 is removed from the solution and 5 being the second highest number in the solution removes every other number in the solution using the (x>y) reaction rule.

**Temporal bonds:** Temporal representation is necessary in modelling context-based environments as it enables the sequencing of interactions and also provides conditional triggering of events and services. C<sub>2</sub>A models absolute temporal constraints by adding a *Boolean* temporal context element to a reaction (catalyst), and models relative temporal constraints by making the reactions sequence non-associative and non-commutative. In the



HOCL encoding, an absolute temporal constraint is modelled as a ‘pair’ (P1:P2), which is a form of pattern P in HOCL grammar. A pair is used such that P1 is a context element or molecule, and P2 is a temporal context element. The pair is used in conjunction with a named gamma reduction rule called ‘temporal’, as shown in Eq. 5,

$$\gamma(P1:P2)temporal.M \quad \text{Eq. 5}$$

For relative temporal bonds (for example, enable reaction A after reaction B) HOCL encoding uses nested solutions, such that, reaction B is enclosed in a solution within the solution containing reaction A. Therefore, conforming to the gamma-reduction principles, a solution containing A can only interact with B’s solution once it has been extracted out of its membrane (solution), and it can only be extracted when either the solution is inert or the reaction A in its composition cannot affect the contents of solution containing B. Eq. 6 summarises this process.

$$\begin{aligned} &<1,2,3,4, \text{ReactionA}=\gamma P \text{ReactionB}.M \\ &\quad <5,6,7,8, \text{ReactionB}=\gamma PC.M> \\ &> \end{aligned} \quad \text{Eq. 6}$$

**Proactive Reactions:** Proactivity is another aspect of a context-based mobile cloud. In context-based mobile clouds, two main interaction modes are considered, i.e. reactive and proactive [30][6]. In the reactive mode, entities of the environment are aware or are enablers of adaptation and awareness and can control it to a certain extent (depending on adaptation logic). In proactive mode the system enables adaptation and awareness on behalf of entities of the environment, and the entities may or may not be aware of, or interested in it.

In future context-based mobile clouds, both the modes are expected to exist in parallel. Whereas it is evident from the discussion above that reactive modelling is very much possible by creating solutions and manipulating their reactions, we now show that using the same approach, autonomous and proactive modelling is feasible as well. In C<sub>2</sub>A we introduced a basic set of context chemistry that consisted of six context-based bonds. We illustrated that whereas any first order algebraic expressions can be modelled as a reaction in C<sub>2</sub>A<sub>HOCL</sub>, every existence of a C<sub>2</sub>A solution must have six named reactions by default and any extension to the set, though possible, is dependent on the user or developer of the system.

$$\begin{aligned} &\text{let associate} = \text{replace } x,y \text{ by } (x,y)_{\text{Association}} \text{ if } x \text{ associates } y \text{ in} \\ &\text{let abstract} = \text{replace } x,y \text{ by } (x,y)_{\text{Abstraction}} \text{ if } x \text{ abstracts } y \text{ in} \\ &\text{let interface} = \text{replace } x,y \text{ by } (x,y)_{\text{Interface}} \text{ if } x \text{ interfaces } y \text{ in} \\ &\text{let temporal} = \text{replace } x,y \text{ by } (x:y) \text{ in} \\ &\text{let proximate} = \text{replace } x,y \text{ by } (x,y)_{\text{Proximity}} \text{ if } x \text{ isInProximityOf } y \text{ in} \\ &\text{let subscribe} = \text{replace } x,y \text{ by } (x,y)_{\text{Subscription}} \text{ if } x \text{ subscribes } y \text{ in} \\ &(<>, \text{associate}, \text{abstract}, \text{interface}, \text{temporal}, \text{proximate}, \text{subscribe}) \quad \text{Eq. 7} \end{aligned}$$

Eq. 7 presents the basic minimal autonomous C<sub>2</sub>A solution. The solution consists of six basic and primary context bonds and an empty solution. In an autonomous mode the data (context elements), is retrieved from the context periodic table, and a solution generates all and any possible reactions between context elements using the primary set of bonds, and outputs an inert solution.

This leads us to the visual representation of a solution, which is done using graphs, called smart space graphs, discussed in the previous section. For example, let us consider an inert solution shown in Eq. 8,

$$\langle (\text{UserA}, \text{GPS})_{\text{Association}}, (\text{UserA}, \text{Time})_{\text{Association}}, (\text{UserA}, \text{Location})_{\text{Subscription}}, (\text{GPS}, \text{Location})_{\text{Interface}} \rangle$$

Eq. 8

The solution in Eq. 8 represents a user ‘UserA’ with four context elements in its CPT: a GPS sensor, a Time source and a subscription to a service ‘Location’. ‘Association’ bonds show the ownership of the GPS and Time context elements to UserA. The ‘Subscription’ bond shows subscription and the ‘Interface’ bond says that the output of GPS is fully or partially consumable by the input of ‘Location’ service. An inert solution serves as an input to the visual representation, which is called a smart space, and can be described as a higher abstraction or a snapshot of context interactions in an environment at a given instance.

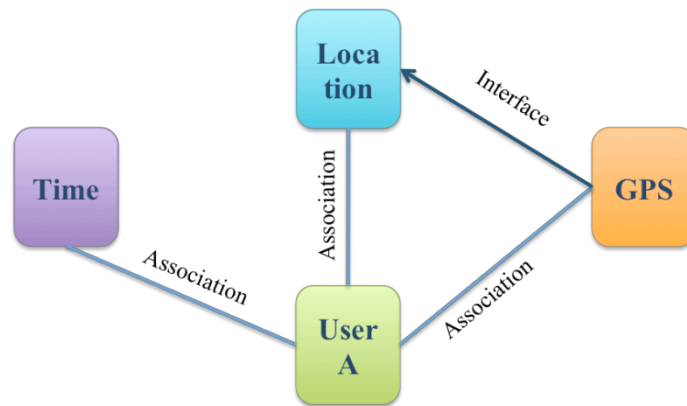


Figure 2 – Smart space graph for solution in Eq. 8

Figure 2 presents the visual representation (smart space) of the snapshot in Eq. 8. Smart space graphs provide a visual representation and interaction model for users and developers. For example, in Figure 2 any update in GPS context element value feeds to the Location service that converts the GPS coordinates to a street address. This interaction was not defined but was dynamically discovered in the CPT of the user ‘UserA’. It is worth mentioning that the nodes are differentiated by their location in the context periodic table such that each column of the context periodic table is represented by a different colour, and the edges are differentiated by colour and texture. The smart space graph serves as an extension of the chemical model and as an input to the HLPNG encoding of  $C_2A$ .

We use the HOCL encoding and smart space graphs to evolve a HLPNG encoding for the system described above. A HLPNG consists of two components and a High Level Petri-net and a graphical representation of the HLPN.

$$HLPN=(P, T, D:\text{Type}, Pre, Post, M_0)$$

A HLPN consists of a set of places  $P$ , a set of transitions  $T$ , a function  $D$  that assigns types to places, Pre/Post functions to assign pre and post conditions for transitions and places and an initial marking for places  $M_0$ . A HLPNG uses HLPN and is shown as,

$$HLPNG=(PNG, Sig, V, H, Type, AN, M_0)$$

It consists of a PNG (which consists of places and transitions), a signature ‘Sig’ that is the boolean algebra used by the HLPNG, a set of variables  $V$  which is sorted by Sig, a many sorted algebra  $H$  that serves as an instantiation of the signature Sig. AN is a pair of annotations that is assigned to each arc and consists of an expression and a Boolean

expression (Expression, Expression<sub>Boolean</sub>). Finally,  $M_0$  is a function built using the many sorted algebra  $H$  that assigns initial markings to places.

HLPNG can be used to model both reactive and proactive solutions, however, to limit the scope of this paper we will only discuss the proactive reactions mode, discussed above. In order to model a  $C_2A$  solution as a HLPNG, we inherit the algebra, data structure and multiset from the HOCL model. Named reactions in HOCL become the operators for the signature  $S$ , and the types of elements in the multiset map to the sorted set  $S$  of signature, whereas, the condition of the  $\gamma$ -reduction maps to the functions  $F$  of the sorted algebra.

The resulting HLPNG can be represented as,

*PNG:Places=Molecules from an inert solution*

*Transitions=Bonds between molecules*

*Signature=S, O*

*S=SmartSpace, Element, Molecule, Compound, Bond, Boolean, List*

*O=Reaction, Add, Remove, Owner, Load*

*Predicates, P=Reliable, Status, Check, Type*

*Functions, F=Reaction, Add, Remove, Owner, Load*

*oStatus=Element, Boolean*

*oReliable=Element.Element, Boolean*

*oCheck=Element.Element.Bond, Boolean*

*oType=Element, String*

*oReaction=Element.Element.Boolean, Molecule*

*oAdd=Element, SmartSpace*

*oRemove=Element, SmartSpace*

*oInit=SmartSpace, SmartSpace*

*oInput= Element, List*

*oOutput=(Element, List)*

Eq. 9

The signature, places and transitions are the basic building blocks of the HLPNG and the remaining components are derived from them.  $H$  is a many sorted algebra that provides an instance of interpretation for the signature 'Sig', such that:

*H=SH, OH*

*for example,*

*HSAMPLE=((GPS, Time, Song, HeatSensor), (Status;Boolean, Reaction;Molecule))*

V is an S-indexed set of variables that are used to annotate arcs originating from places and transitions. Variables and constants are derived similarly as functions and predicates in Eq. 9 however their input sorts are set as empty or null.

Similarly, each context element has a value at any given instance or they are assumed offline. The values at the time of creation of smart space are mapped to the initial marking.

$$M_0 = \{\text{Initial values of context elements}\}$$

Lastly, using terms generated from the sorted algebra and sorted variables, markings for the arcs are created. Figure 3 exemplifies the steps above that uses the solution from Eq. 8.

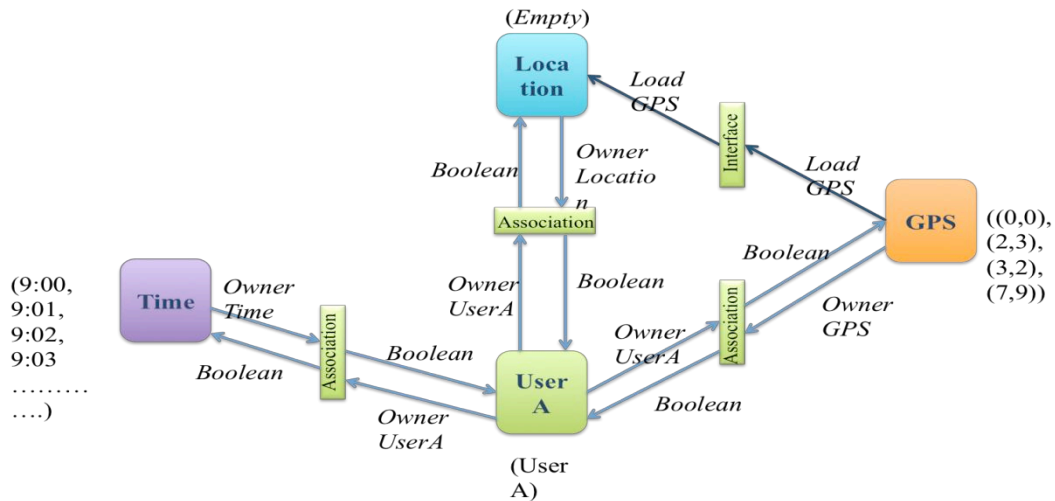
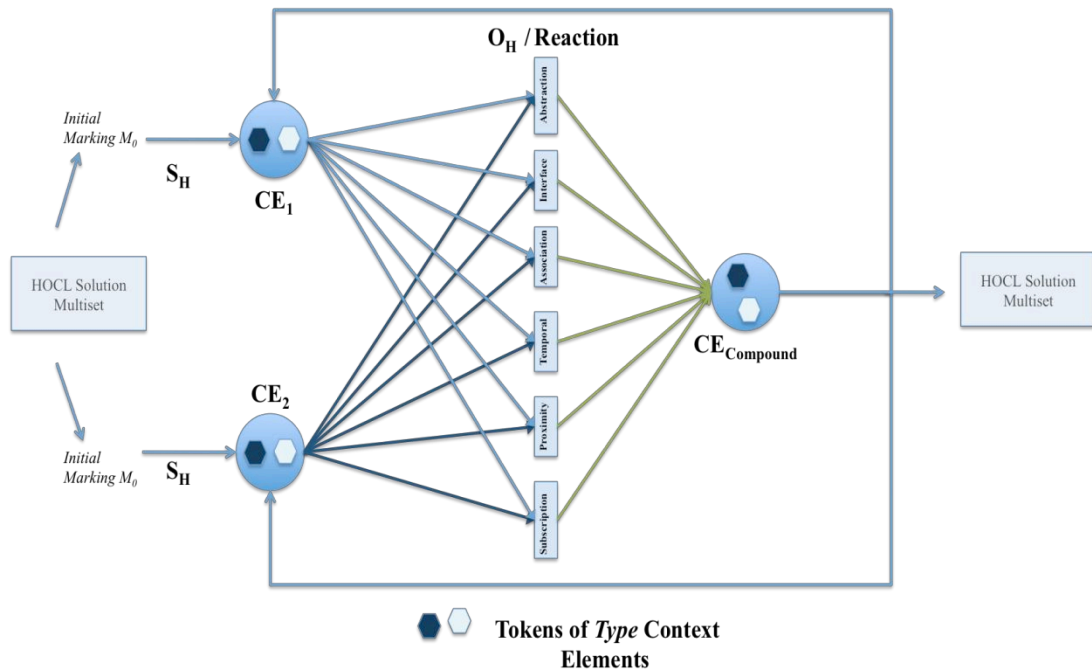


Figure 3 – HLPNG for smart space shown in Figure

It can be observed in Figure 3 that the markings are marked in brackets such that the first element of the set represents the initial marking and then onwards every element occurs sequentially in linear time. The markings on the arcs are generated using terms created from variables and operators defined by the algebra (see Eq. 9).



Tokens of Type Context Elements

Figure 4 – Basic proactive HLPNG for a C<sub>2</sub>A solution

The output of the proactive HLPNG shown in Figure 4 is a multiset (solution) of context molecules and compounds that itself can be modelled as an instance of HLPNG as was per Figure 4.

#### 4. Scenario-based Evaluation

In this section a scenario-based evaluation of the proposed HOCL-Petri-net encoding of C<sub>2</sub>A is presented. A controlled mobile cloud of users, artefacts and context sources is considered as the basis for modelling context and interactions.

**Scenario:** Figure 5 presents an evaluation storyboard to be used for the evaluation. Using the storyboard, three of the entities involved, User E, User F and Mall are modelled. The black rectangular marking (both solid and dashed) specifies the scope of an encompassing entity.

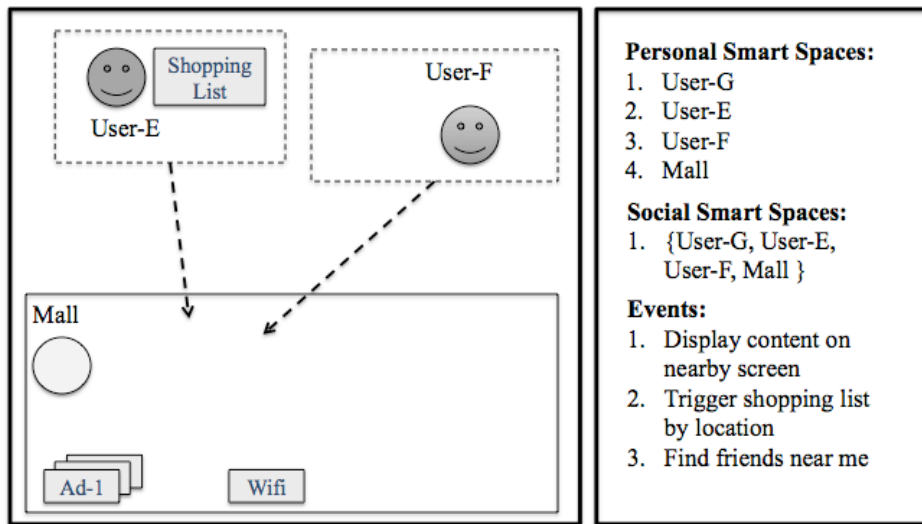


Figure 5 – Sample scenario for the experiments

**C<sub>2</sub>A Model for the Scenario:** The grammar and structure from Eq. 5, 8 and 9 are used to model the CE for the 3 entities involved. The Petri-net model from Figure 4 is used to generate a smart space solution and a graph, similar to Eq. 8 and Figure 3. The resulting petri-net model of the smart space is executed manually with pre-defined context values, and the output of the petri-net execution is analysed in terms of suitability of the response, alternate execution flows, and reachability of the petri-net graph.

Where [1]	When [2]	How [3]	What [4]	What' [5]	Who [6]
Mall	System Clock	Display	Wn-1 <sub>4</sub> Services	Song A	User E
		Wifi	Wn <sub>4</sub> Tasks	Movie B	FamilyList
				Shopping List	
			Wn-1 <sub>2</sub> Services	Find Screens	FindFamily
			Wn <sub>4</sub> Tasks	Shopping List	
<b>User E</b>					
Where [1]	When [2]	How [3]	What [4]	What' [5]	Who [6]
(X,Y)	System Clock	GPS	Wn-1 <sub>4</sub> Services	SMS	User F
Address		Display	Wn <sub>4</sub> Tasks	MMS	
			Wn-1 <sub>2</sub> Services	Events Around	Friends
			Wn <sub>4</sub> Tasks	FallAlarm	
<b>User F</b>					

Where [1]	When [2]	How [3]	What [4]	What' [5]	Who [6]
RFID Tag	System Clock	Wifi	Sale	Ad-1	Mall
	Now		Closed	Ad-2	Top Customers
	Tomorrow		Wn-1 <sub>4</sub> Services	Coupon-1	
			Wn <sub>4</sub> Tasks		
			Wn-1 <sub>1</sub> Services	NearMe	
			Wn <sub>1</sub> Tasks		

**Mall**

Figure 6 – Context Periodic Tables for entities on the storyboard (a) User E (b) User F (c) Mall

From Figure 6, three individual personal smart space scopes are identified and their corresponding CPTs are shown in Figure 6. Now that the storyboard and CPTs for the entities involved in the scenario have been illustrated, the HOCL encodings of the CPT for User E is presented in Eq. 10 and is further used for modelling CEs and reactions. As the processes for modelling CPTs and CEs are similar, the CE representations of only 3 CEs are shown to avoid repetition.

$$\begin{aligned}
 \text{Solution}_{\text{UserE}} = [ & \text{UserE}, \text{FamilyList}, \text{SongA}, \text{MovieB}, \\
 & \text{ShoppingList}, \text{Display}, \text{Wifi}, \text{SystemClock}, \\
 & \text{FindScreens}, \text{FindFamily}, \text{ShoppingList}, \text{Mall}]
 \end{aligned}
 \tag{Eq. 10}$$

The CPT for User-E is composed of 12 CEs, Figure 7 presents 3 CEs out of 12 to show the representation of three different types of CEs and it is assumed that the remaining CEs are modelled in a similar manner. In Figure 7, SongA is a CE of type WHAT', it has no input and it outputs a single data structure called 'content'. The CE SongA is available, does not belong to any category and is not a composition. Alternatively, Wifi CE has an input of type 'hardware' and provides a 'connection' as an output and is of type HOW. Finally, ShoppingList CE is of type Task Element (TE) and is a compound (composite context). This is composed of four atomic CEs (GPS, Mall, ShoppingList and Display), and it takes as input GPS-CE and outputs data of type 'content'. The internal structure of the TE may vary and can be derived from the petri-net representation in Figure 4.

SongA		Wifi	
Physical Properties	Chemical Properties	Physical Properties	Chemical Properties
Input = {none}	Output = {content}	Input = {hardware}	Output = {connection}
Type = {5}	Reliability = {1}	Type = {3}	Reliability = {2}
Title = SongA	Tags = {audio}	Title = Wifi	Tags = {wireless}
Category = $\phi$	Status = {1}	Category = $\phi$	Status = {1}
Boolean = {true}	Composition = {}	Boolean = {true}	Composition = {}
ShoppingList			

Physical Properties	Chemical Properties
Input = {GPS}	Output = {content}
Type = {4}	Reliability = {}
Title = ShoppingList	Tags = {}
Category = $\phi$	Status = {1}
Boolean = {true}	Composition = {GPS, Mall, ShoppingList, Display}

Figure 7 – CE representation for SongA, Wifi and ShoppingList CEs

Now that CE and CPT structures have been shown the entity solutions are defined for UserF and Mall:

$$\begin{aligned}
 \text{Solution}_{UserF} = & [(X,Y), Address, SystemClock, GPS, \\
 & Display, SMS, MMS, UserF, \\
 & EvenetsAround, Friends, FallAlarm]
 \end{aligned}
 \tag{Eq. 11}$$

where FallAlarm = replace *Dial* with (*Dial, Emergency*) if (*accelerometer* > 100)

where *Dial* and *accelerometer* are CEs and (*Dial, emergency*) is a compound based on interface bond.

$$\begin{aligned}
 \text{Solution}_{Mall} = & [RFIDTag, SystemClock, Now, Tomorrow, \\
 & Wifi, Sale, Closed, Ad-1, Ad-2, Coupon-1, \\
 & Mall, TopCustomers, NearMe]
 \end{aligned}
 \tag{Eq. 12}$$

where *NearMe* is a service that *Mall* is subscribed to and it notifies Mall of people near its location.

An overall solution would look like:

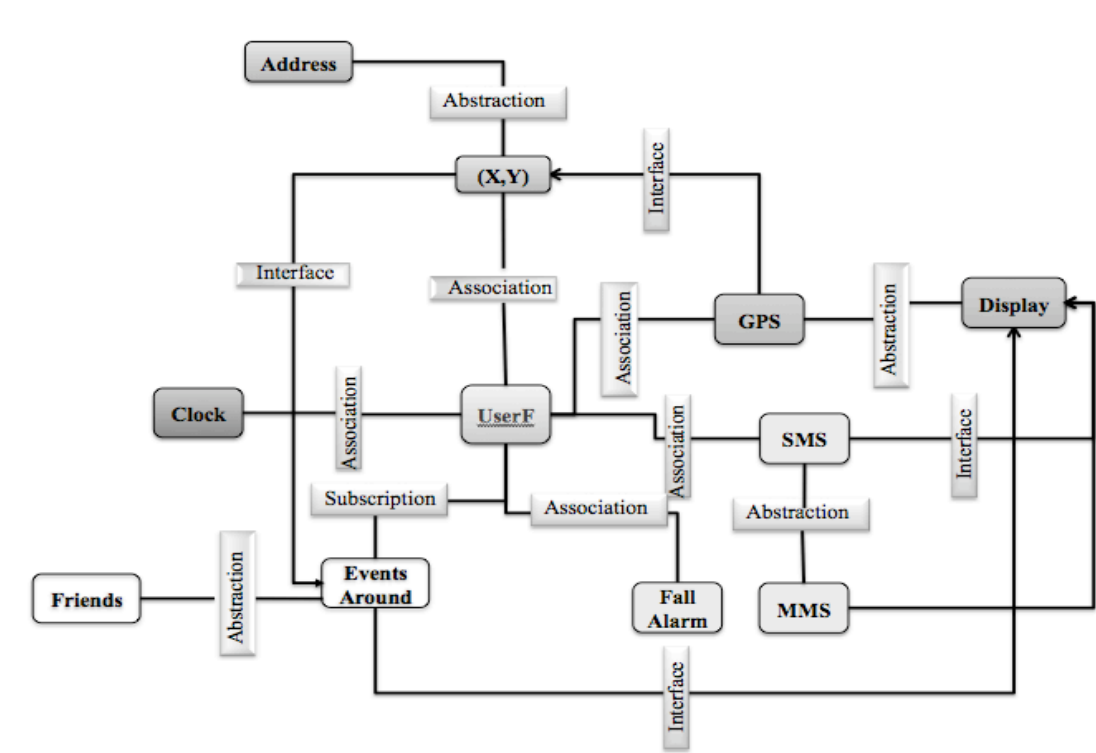
$$\begin{aligned}
 \text{Solution} = & [\text{Solution}_{UserE}, \text{Solution}_{UserF}, \text{Solution}_{Mall}] \\
 \Rightarrow & [UserE, FamilyList, SongA, MovieB, Display, \\
 & Wifi, SystemClock, FindScreens, FindFamily, ShoppingList, \dots, \\
 & [(X,Y), Address, SystemClock, GPS, Display, SMS, MMS, UserF, \\
 & EvenetsAround, Friends, FallAlarm, \dots, RFIDTag, \\
 & SystemClock, Now, Tomorrow, Wifi, Sale, Closed, \\
 & Ad-1, Ad-2, Coupon-1, Mall, TopCustomers, NearMe]
 \end{aligned}
 \tag{Eq. 13}$$

Eq. 13 presents the collective solution of three entities involved in the scenario. In the proactive mode, if we feed individual solutions to the petri-net model shown in Figure 4 we get the output as another petri-net that represents a Personal Smart space (PSs). The resulting petri-net can be executed based on the values of context sources at a given time. Similarly, if the same petri-net gets as input the combined solution of entities as sampled in Eq. 13 then the output is a Social Smart space (SSs). In this section we model one instance of PSs using Petri-nets. Consider the solution for User F described in Eq. 11. It consists of 11 Context Elements (CEs) out of which 8 are basic CEs, 2 are of type Service Element (SE) and 1 is of type Task Element (TE).

	Context Elements	Bond
1	UserF, SMS	Association
2	UserF, GPS	Association
3	UserF, SystemClock	Association
4	UserF, (X,Y)	Association
5	UserF, EventsAround	Subscription
6	UserF, FallAlarm	Association
7	SMS, MMS	Abstraction
8	GPS, Display	Abstraction
9	(X,Y), Address	Abstraction
10	FallAlarm, Accelerometer, Dial, EmergencyContact, Display	Interface
11	(X,Y), GPS	Interface
12	SMS, MMS, Display	Interface
13	EventsAround, (X,Y)	Interface

Table 5 – List of reactions in User-F's CPT

**Petri-net Model for the Scenario:** Using the Petri-net model from Figure 4, Table 5 shows a subset of reactions that can take place considering the CPT of User-F. It is important again to mention here that the reactions listed above are not the only possible reactions. Even for similar resulting compounds, the CEs can randomly interact and may yield a different path of reactions to the same or even a different output compound. For example, the reactions listed in 10 and 12 in Table 5 could be broken down. Similarly, reaction 10 is incomplete as there is no accelerometer, emergency contact or dial CE in the CPT. As the CPT is a dynamic data structure, CEs can register online and offline depending on their status, which is derived by their respective handler.





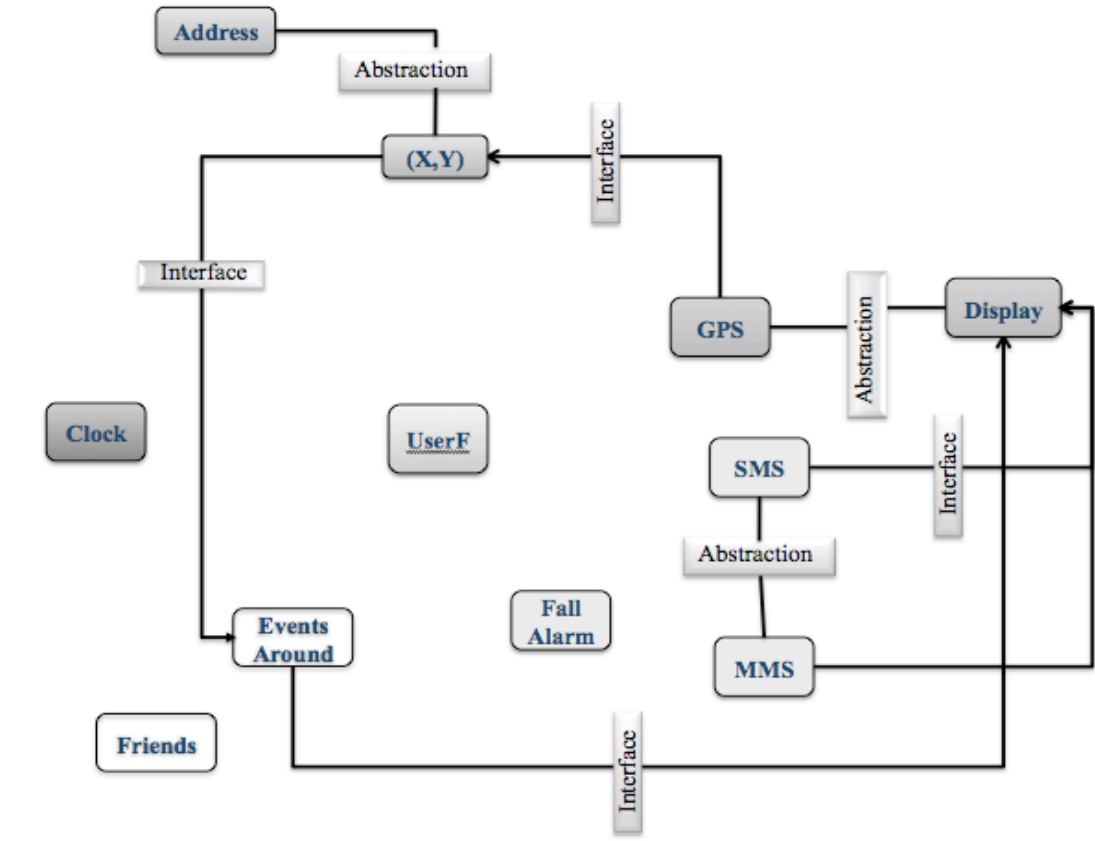


Figure 8 – Petri-net model for the PSs of User-F (a) Complete (b) Filtered

Figure 8 shows the Petri-net model for reactions listed in Table 5, which were generated using the CPT shown in Figure (c). The Petri-net representation in Figure 8 can be tested and analysed for different sets of input tokens. For example, if GPS-CE is turned on and sends an update to the (X,Y)-CE it will further trigger the EventsAround-SE, which may or may not find a valid output to be sent to the display. Similarly, SMS and MMS both are bonded with display, in an instance, where the device gets an SMS and requests an exclusive hold of display, it will depend on the device already being involved in a bond or on the one, which is still unused.

**C<sub>2</sub>A deduction to form a SmartSpace:** So far, the proactive mode of C<sub>2</sub>A was modelled using the formal encoding, however, reactive awareness and user created mashups are an important aspect of C<sub>2</sub>A. The process to compose reactive awareness is carried out using the  $\gamma$ -reductions of HOCL and many-sorted algebra of Petri-nets (Eq. 9). Let us consider the compound ‘ShoppingList’ shown in CPT (Figure 7) to analyse if we can model all aspects of the scenario using the encodings derived above. Let us assume that the logic behind creation of this mashup is that ‘if the user is in a shopping mall called ‘Mall’, shopping list content should be activated and displayed’. The overall solution for the mashup would be as follows,

*Solution = (GPS, ShoppingList, Display, Mall)*

The CE element ‘Mall’ could be manually created by providing GPS coordinates of the ‘Mall’, or the user can create a CE ‘Mall’ if he is in the mall and uses an ‘abstraction’ bond between the current GPS coordinates and a new CE of type ‘WHERE’ and title ‘Mall’, such as,

- ➔ // Check existence of abstraction bond between GPS and Mall
- ➔ Bool check = (GPS.Mall.Abstraction, *Boolean*)

```

1      → // Create abstraction bond
2      → Molecule Location_Map = (GPS.Mall.check, Molecule)
3
4      → // Check if shopping list exists
5      → Bool SL_Alive = (ShoppingList, Status)
6
7      → // Check is the solution has any display
8      → Bool display = (Display, Status)
9
10     → // Create a trigger using abstraction molecule and existence of display
11     → Bool trigger = (LocationMap.SL_Alive.display, Boolean)
12
13
14     → // Execute trigger with an interface bond between list and display
15     → Bool execute = (ShoppingList.Display.trigger, Boolean)
16
17
18     → // Create a smart space (If needed)
19     → SmartSpace SS = Init(SS, SmartSpace)
20
21     → // Add elements and compound modelled to the smart space
22     → SS = Add(ShoppingList.Location_Map.Display, SS)
23

```

The petri-net for this mashup is shown in Figure 9. The scenario-based experiment has shown that the formal encoding of C<sub>2</sub>A, using HOCL and petri-nets, can be used for modelling, validation, analysis and visualisation of context-based pervasive scenarios. The encoding with the many-sorted algebra enables expression of rich logic into services and mashups, and the branching and nesting capabilities of chemical computing and petri-nets provides an environment for the modelling of finite-but-large application behavioural possibilities into mobile cloud-based situations. The applications of the chemical context model and its formal encoding can be separated in two aspects: awareness and adaptation modelling (in the context of applications and services) and validation of the designed awareness. Validation methods such as reachability and walks can be used to validate the usage of a given context source in a given mobile cloud. Trends and patterns can be used to analyse the most active or used context in a given environment. By having a formal representation of a mobile cloud, context sources such as sensors and multimedia, can be intelligently placed in their target environments for maximum reachability and liveliness. Graph analysis methods such as dominating sets can be used on the petri-net instance graphs to identify most lively context source in an environment. In the following section we present the results based on the higher-level simulation-based implementation of the proposed chemical formal model.

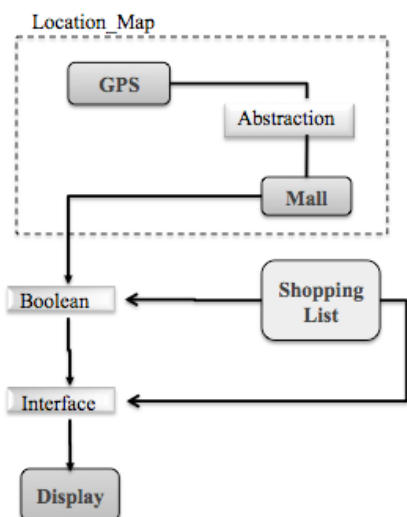


Figure 9 – ShoppingList mashup

## 5. Implementation and Results

Although the focus of this paper is to lay emphasis on the formal base of chemical computation models, in this section we test a high-level implementation of the proposed formal model in a Java-based simulator [1]. Garzonis [41] has highlighted the complexities involved in implementing experimental setups to test context aware solutions, summarised a survey on the use of lab-based and field-based trials and proposed a combination of the two for completeness of results.

The setup of simulator is discussed at length in [1]; the datasets used in these experiments are generated using a pre-defined pool of simulated context and information sources in the simulator. For every iteration the simulator picks a collection of CEs for CPT from the pool based on the column based configuration, for example, it can be specified in the experiment configuration that for this experiment distribute equal numbers of CEs across all columns of CPT and more.

Context bonds serve as the relational links between context elements. For statistical and observational perspectives in the following experiment distribution of bonds in different scenarios is observed. In the first instance the graph in Figure 10 shows the change in total number of bonds in a Ss with increasing number of CEs. It is observed that the variation is towards being exponential and this shows the large variety of interactions that can be realised, as each bond represents a possible interaction.

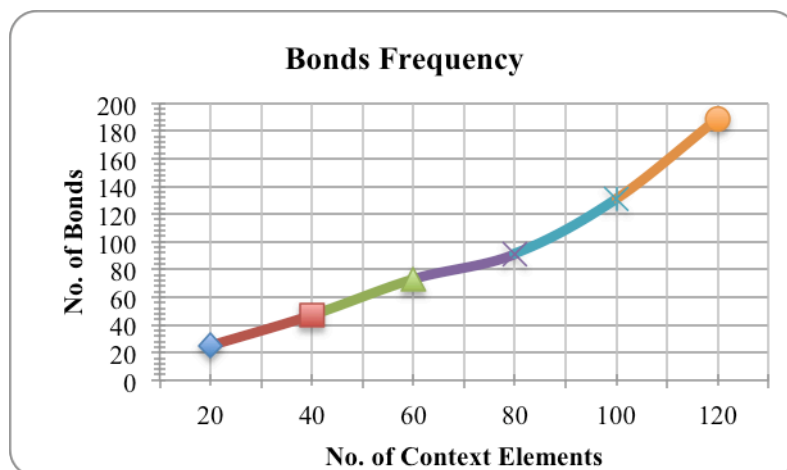


Figure 10 – Bonds frequency chart in a Ss

Building on the observation that context bonds tend to grow exponentially with increasing context elements, the next experiment focussed on exploring further the distribution of bonds within a smart space graph. It was expected that the minor deviation in ratio of bonds (between 80 and 100 CEs in Figure 10) might have occurred because of the distribution of bonds type within a smart space. The results of this experiment are shown in Figure 11. It was observed that the distribution pattern of bonds within a space tends to stay consistent, with minor variation (mainly in abstraction and interface bonds, see Figure 11).

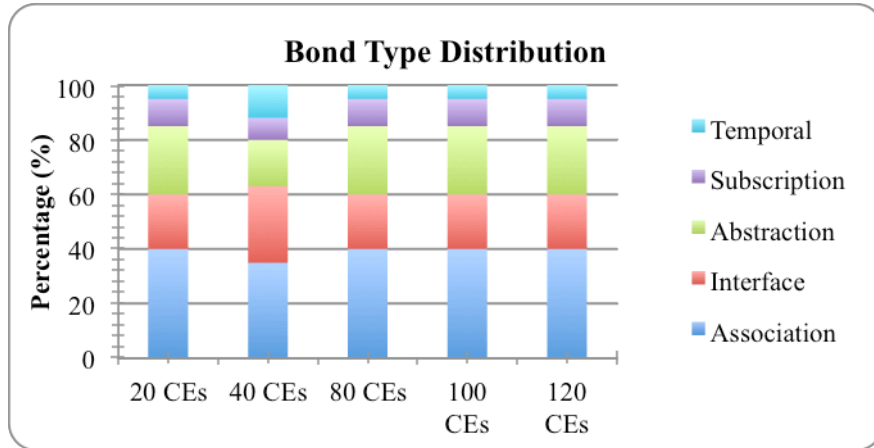


Figure 11 – Distribution of context bonds within a PSS

Further exploration suggested that this was because of the organisational structure of CPT that enabled similar distribution ratios for CEs though increased but selected on same criteria. In this case it was uniform distribution across the CPT. Association bonds show the ownership and association of a CE to a user or device, therefore, in any scenario they appear to be the majority of the bond distribution ratio. Interface bonds typically form between content (What') and actuator elements (How), therefore, a CPT with these two types of elements have second highest ratio of interface bonds. Similarly, any CPT with higher number of elements in the same column type will enable abstraction bonds. Whereas, subscription and temporal bonds are mostly user defined and are not predictable, which shows why they appear to have lowest ratio in simulation scenarios.

In following experiments two applications modes of the C<sub>2</sub>A model; service composition and dominating set identification are studied. Figure 12 presents the results for service composition time consumption using two atomic services structures, central and hierarchical, for a randomly generated service composition request but increasing number of potential services to search from. It was observed that the resulting time was dependent on whether hierarchical structure found a match in the best-case scenario (first level) or worst-case scenario (n<sup>th</sup> level). Therefore the values in the graph show the mean for a set of 5 best-case and 5 worst-case runs. For example, for the given experiment, GPS coordinates and 'News' content elements were passed in the request signature and the service repositories were preloaded with a SE that converts GPS coordinates to 'city name' and another SE that uses 'city name' and provides news for that city.

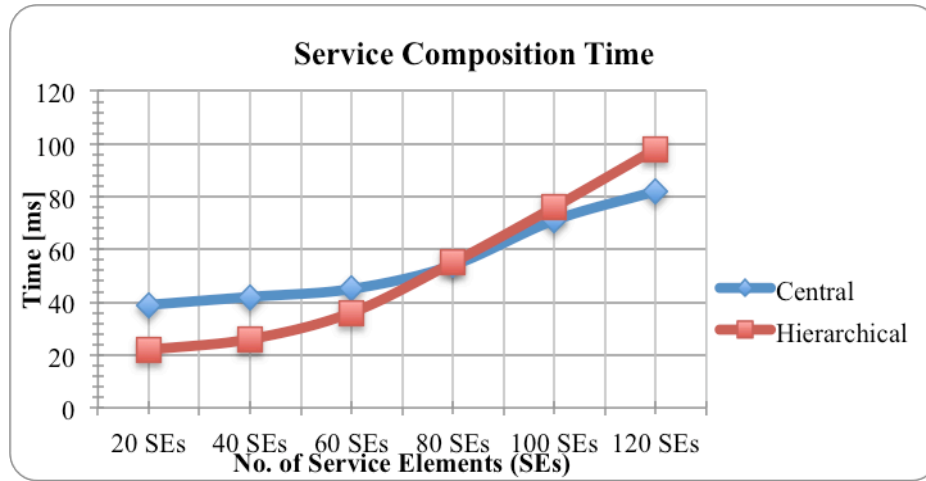


Figure 12 – Service composition time consumption

In this experiment for hierarchical structure, two levels were used. It was observed that the hierarchical structure offer performance benefits till a certain threshold, in this case approximately 60-75 SEs, but onwards the performance gets slower than the central repository model due to messaging overheads. However, it is still an indicator that various search and organisation models can be tested for hierarchical storage and placement structure for service repositories. However, these ideas are not further investigated in this paper but are considered as a possible direction for future work.

In the next experiment, second application of smart spaces, identification of dominating set of nodes in a smart space graph, is analysed. In this experiment, shown in Figure 13, dominating sets were identified for randomly generated smart space graphs. As previously stated failure instances where no dominating set could be found were ignored. The result was linear computational time indicating that computational time was proportional to node degree distribution of the graph, i.e. the node connection complexity.

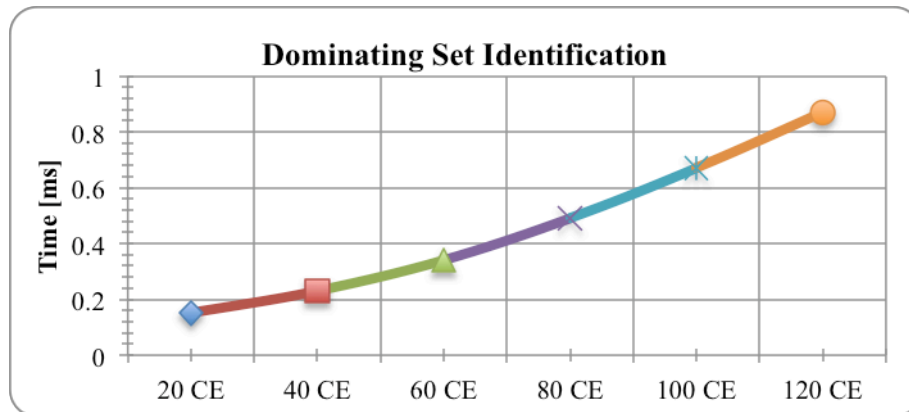


Figure 13 – Dominating set identification time consumption

To confirm this observation the same experiment was re-run and node-degree distribution of the graphs were observed, shown in Figure 14. It was found that the Ss graphs are by nature scale free and conform to the power law distribution, which is consistent with previous research into graphical representations of social networks [43]. The reason for this distribution pattern lies in the generation algorithm which is developed such that ‘association’ nodes, representing the association (owner-owned) between CEs are dominant as every CE is owned by a parent CE, similarly, abstraction bonds become prominent within a CPT column however as the CPT grows and further types of bonds are added to the graph the new

additions are CE specific such as temporal bonds, interface bonds and therefore do not occur in clusters.

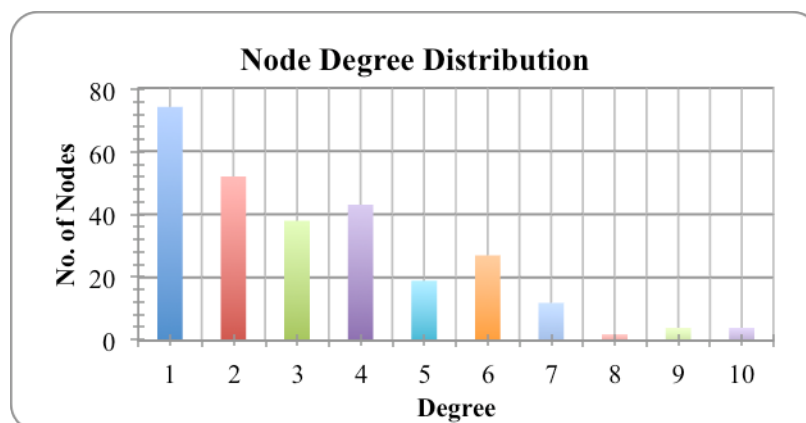


Figure 14 – Node-Degree distribution of Ss graphs

So far we have tested whether the chemical model implementations works and if it does what are the performance indicators. In the following experiments, we test the visualisation and modelling aspects of the chemical model using scenarios/storylines.

#### a. Storyline-based Analysis

In [1] we have presented storyline-based experiments in detail. Such experiments have contributed towards analysing creation and existence of personal and social smart spaces in parallel. They have also contributed towards presenting the visual modelling aspect of the proposed formal model and hint at the possibilities that visual modelling can have on concepts such as ‘Internet of Things’ and ‘Context Awareness’ in terms of interaction and orchestration of personal and social spaces of an entity or a user. In this paper we build on the previous findings and extend the analysis towards the qualitative dimensions of the chemical context model.

#### b. Service Selection and Orchestration

Service selection and orchestration is proposed as an application area of the chemical model. Services are part of the CPT and appear as Service Elements (SE). SEs are a type of CE with an added signature that is used for selection and matching of a service to a particular context or situation, where, context is a single CE and a situation is a collection of CEs arranged by CPT columns to depict a situation.

Navigation Service Element						
CPT Column	Where	When	How	What	What’	Who
Categories	Location	*	*	*	*	*
Value	X <sub>1</sub> , X <sub>2</sub>	*	3G/Wifi	Travel/Search	*	U <sub>1</sub>

Table 6 – Service Element Signature Example, Navigation Service

Table 6, shows the *signature* part of a sample *Navigation* SE. The example suggests that this service is a candidate for a situation involving exactly two *Where* elements of category *location*, having at least one *How* element of any category with either a 3G or Wifi context element, involving *What* elements of any category having values Travel or Search and having at least one *Who* element of any category.

A service search and selection request is primarily a multiset containing CEs from a smart space graph or context periodic table. The experiments were setup such that a set of 15 services was broken down into sub-parts (and represented as service signatures) such that

combining the parts using the logical operators (AND, OR and IF) would result in the initial composite service. The number of combined sub-parts was kept to 100 by topping up the sub parts with randomly generated service signatures using the S3 simulator. The resulting collection of 100 services was randomly mixed together and distributed in two levels to depict services at gateways and at central tuple space, see Table 7.

	Service	Sub-Services
1	Search News for Location	Search Location for GPS Search News for Location
2	Get Content by Language	Search content Translate content by language Display Content
3	Book Journey	Find Hotel by Preferences Find Taxi by Language
4	Share content	Find content Find sharing medium Find sharing display
5	Find social network by location and interest	Find social network by location Find social network by interest
6	Share a shopping list	Create a shopping list Share a shopping list in a social network Update shopping list
7	Create social network polls	Create poll Create social network and share poll
8	Get content by device settings	Get content Transcode content for device settings
9	Search Weather by Location	Get location Get weather for location
10	Find me	Find a location reference Update location data
11	Detect Activity	Get onboard sensors Predict activity
12	Detect Activity	Find social network Inherit activity
13	Patient monitoring	Create hospital social network Create emergency activities
14	Personal Assistant	Get location Get information for location
15	Preserve Energy	Detect activity Trigger household devices
	<b>15 Services</b>	<b>33 Sub-parts</b>

Table 7 – List of services for service composition experiments

The experiments in this section were focussed on examining the applicability of C<sub>2</sub>A model for service composition in terms of its accuracy and success ratios. Exemplary services, listed in Table 7, were used for analysis. In the first experiment it was observed that with increasing number of CEs in a service composition request, how many requests were returned with at least one complete composition template and how many requests resulted in failure. Each service composition request with n number of CEs (where 1 < n < 20) the experiment was run five times. SEs were pre-loaded in the simulator and a PSs is generated with growing CEs for each PSs, CEs are sorted by node-degree count and the highest ranked CE is used as service request, for example, for the first set (first experiment in Figure 15 where service composition request size is 1) where service request size is one, only the highest ranked CE is used and so

on. For this experiment only the complete composition results were considered, where, a complete composition result is a set of services that collectively match to all CEs in the initial request. Lastly, each request size was executed until five success or failures were achieved.

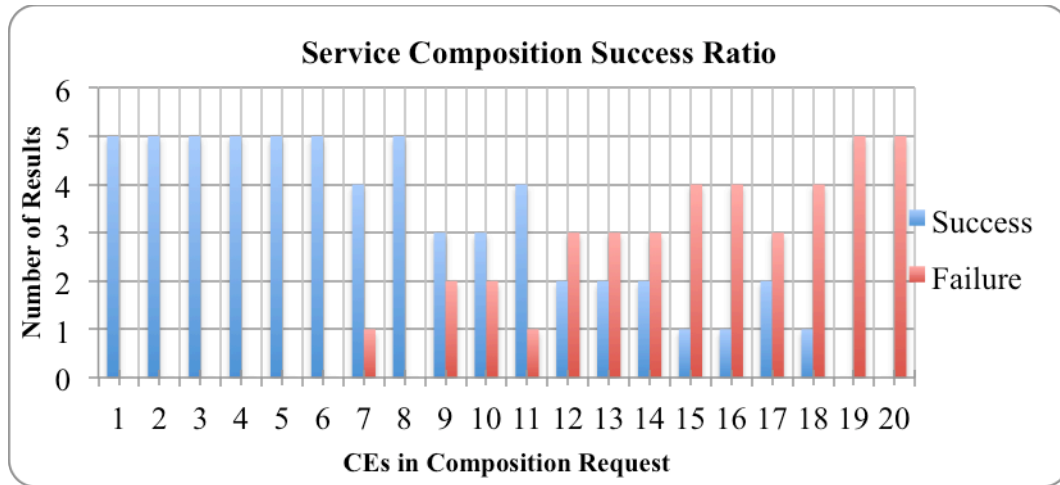


Figure 15 – Service composition success ratio

Figure 15 presents the results of the above-mentioned experiment. It is observed that ignoring the partially completed responses the success ratio decreases with increasing composition request size. This is because with a large composition request the probability of conflicts rises. Therefore, in such scenarios, although there will be increasing partially satisfied completions it is difficult to find an end-to-end composition satisfying the complete request parameters without involving a decision making entity. However, it is important to point out that these experiments were conducted using basic string and interface matching and the efficiency of service composition application of the chemical model can be further investigated by enhanced semantics and ontologies involved in the process of selection.

Figure 15 and Table 7 show the working for a single instance of the above experiment where request size is 5 and the output is a composition success. Table 7 shows how during the search process sub-services were mapped to the request signature based on the values of CEs and Figure 16 how the sub-services were linked together to form a composite service. It is also worth noticing that semantically 'GetLocation' component of the composition is not useful for the composition as the PSs already has a GPS CE, however, being part of the mapping procedure it is added to the service graph. At the moment no filtering or pruning mechanisms are adopted to clean a composition response.



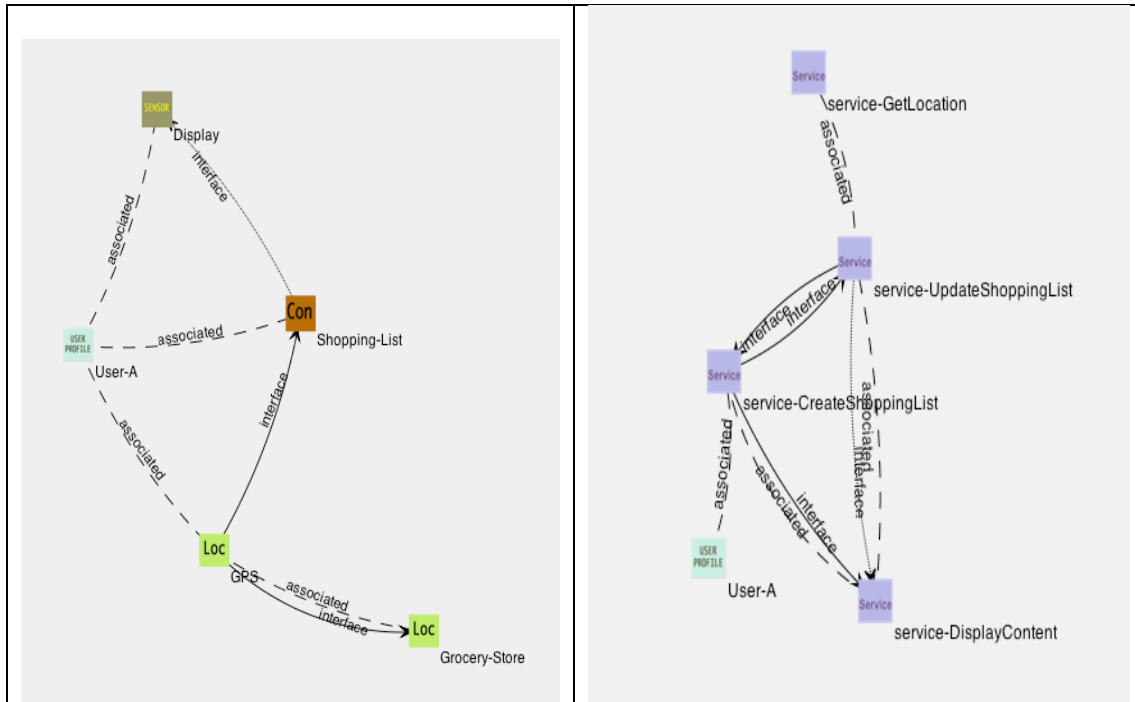


Figure 16 – Service composition scenario (a) Personal Smart Space (b) Service composition response

Where	When	How	What	What'	Who
GPS Grocery-Store		Display		Shopping-List	User-A
				<i>CreateShoppingList</i>	
<i>GetLocation</i>					
		<i>Display Content</i>			
				<i>UpdateShoppingList</i>	

Table 8 – Service signature mapping to composition request

Next, Figure 17 shows the affect on combined compositions, complete and incomplete, for composition requests same as Figure 15. It is observed that whereas complete composition declined with increasing request size the overall response size (where size is measured as number of CEs in a request or number of SEs in a response) increased with increasing request size, which means, that the rise is in the number of partial compositions. This suggests that the process for creating sub-services and composition requests needs more investigation as this can affect the overall performance of the composition process. For example, sub-services doing only one task will increase the number of sub-services in any experiment which increases the search space also increases number of incomplete compositions and also conflicts in a scenario. Further analysis of this aspect is part of future directions.

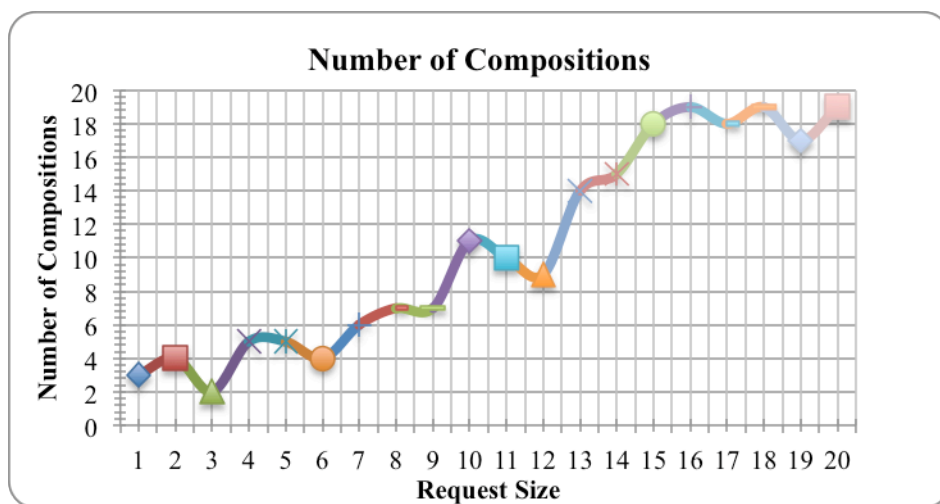


Figure 17 – Number of compositions (complete and incomplete) for growing request size

The experiments in this section support that the chemical computing model with its graphical representation is capable of composing and orchestrating services spontaneously based on a given snapshot of the chemical solution, in this case, the CPT and SS graph.

## 6. Conclusion and Future Directions

Mobile clouds hold great potential to upgrade the richness, interactivity and detail of services and social/personal computing of the future. However, the road to realising that potential requires exploring unconventional methods and approaches to resolve resource management and service creation challenges that the large set of dynamic and growing data that mobile clouds offer. Recently, nature-inspired computation models have been a prominent domain of research in systems that require adaptable and intelligent interfaces and interactions. Whereas human and bio-inspired approaches have been the core focus of research, it can be contested that chemical reactions and chemistry offer a natural metaphorical mapping to the requirements of contextual modelling in mobile clouds. Chemical computing offers promising characteristics and concepts in terms of representing concurrency, dynamic behaviour and scalable architectures for future mobile clouds. The  $C_2A$  model built on chemical principles provides a clear separation of information and logic in CEs and reactions, thereby improving scalability and making the behaviour dynamic, real-time and autonomous. The formal encoding discussed in this article validates its theoretical assumptions and principles.

However, chemical computing is relatively recent as compared to other nature-inspired computation models and needs further investigation in realising chemistry concepts that can benefit a computational model built upon the chemical reactions principles. Ions, membranes, energies of bonds and other related concepts can enable realisation of an autonomous and self managing computation model where information is separated from logic and their interaction generates events and triggers. Similarly, new and improved visualisation models can be investigated that reuse the concepts of shapes, links and graphs and attempt to evolve a standard, interactive and dynamic visualisation model for dynamic information and environments. Moreover, service composition, which was studied as an application of  $C_2A$  needs further investigation into reaction-defining-algorithms, inference engines, ontologies and other relevant methods which can evolve and extend the proposed service composition application domain. Another relevant direction is the storage and retrieval mechanisms of CEs and SEs in the physical setup of mobile and other networks, in this paper we tested minimal central and hierarchical models but as suggested it needs further context-aware investigation, for example, different storage and organisation methods can suit different underlying physical networks and can be used accordingly and if that works how to bridge

communication between two different approaches and so on. All of these issues present interesting directions in which C<sub>2</sub>A can be extended.

## 7. References

- [1] A. Ikram, A. Anjum, R. Hill, N. Antonopoulos, L. Liu and S. Sotiriadis, "Approaching the Internet of Things (IoT): A Modelling, Analysis and Abstraction Framework", *Concurrency and Computation: Practice & Experience*, ISSN: 1532-0634, DOI: 10.1002/cpe.3131, 2013
- [2] J. P. Banâtre, P. Fradet, and Y. Radenac, "Principles of chemical programming," *Electronic Notes in Theoretical Computer Science*, vol. 124, no. 1, pp. 133–147, 2005
- [3] G. Berry and G. Boudol, "The chemical abstract machine," *Proceedings of the 17th ACM SIGPLAN-SIGACT symposium on Principles of programming languages - POPL '90*, pp. 81–94, 1990
- [4] P. Dittrich, J. Ziegler, and W. Banzhaf, "Artificial chemistries--a review," *Artificial life*, vol. 7, no. 3, pp. 225–75, Jan. 2001
- [5] C. Wang and J.L. Pazat, "Using Chemical Metaphor to Express Workflow and Service Orchestration," 2010, pp. 1504-1511
- [6] A. Ikram, N. Baker, M. Knappmeyer and R. Tönjes, "A Chemical Model to enable Context Awareness", 4th IFIP International Conference on New Technologies, Mobility and Security, Paris, 2010
- [7] A. Ikram and N. Baker, "Chemistry for Context Awareness (C<sub>2</sub>A)", in 7<sup>th</sup> International Conference on Intelligent Environments (IE), Nottingham, U.K, July 26<sup>th</sup> – 28<sup>th</sup>, 2011.
- [8] A. Ikram, N. Baker, M. Knappmeyer, R. Tönjes and E. Reetz, "An Artificial Chemistry based Framework for Personal and Social Context Aware Smart Spaces", in 7<sup>th</sup> International Wireless Communications and Mobile Computing Conference (IWCMC), Turkey, 2011
- [9] J.P. Banâtre, P. Fradet, and Y. Radenac, "A generalized higher-order chemical computation model," *Electronic Notes in Theoretical Computer Science*, vol. 135, Mar. 2006, pp.3–13
- [10] J.P. Banâtre, P. Fradet, and Y. Radenac, "Programming self-organizing systems with the higher-order chemical language," *International Journal of Unconventional Computing*, vol. 3, 2007, p. 161
- [11] H. Yoon and W. Woo, "CAMAR Mashup: Empowering End-user Participation in U-VR Environment," 2009 International Symposium on Ubiquitous Virtual Reality, 2009, pp.33-36
- [12] IPCITY (FP-2004-27571) Interaction and Presence in Urban Environments, ([http://www.ipcity.eu/?page\\_id=13](http://www.ipcity.eu/?page_id=13), Accessed: 01/01/2011)
- [13] IPERG (FP-2006 - 004457) Integrated Project on Pervasive Gaming
- [14] Cmobile, EU FP6 Project ([ftp://ftp.cordis.europa.eu/pub/.../20080416-17-c-mobile\\_en.pdf](ftp://ftp.cordis.europa.eu/pub/.../20080416-17-c-mobile_en.pdf))
- [15] C-Cast, Context Casting, EU FP-7 Project (<http://www.ict-ccast.eu/>)
- [16] N. Ibrahim, F.L. Mouël and U.D. Lyon "A Survey on Service Composition Middleware in Pervasive Environments," *Journal of Computer Science*. 2009 1, pp.1-12
- [17] S. Dobson, F. Zambonelli, S. Denazis, et al., "A survey of autonomic communications." *ACM Transactions on Autonomous and Adaptive Systems*. 2006, 1(2): pp. 223-259 Available at: <http://portal.acm.org/citation.cfm?doid=1186778.1186782>
- [18] M. Baldauf, S. Dustdar, and F. Rosenberg, "A survey on context-aware systems," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 2, 2007, pp. 263-277
- [19] G.M. Kapitsaki, G.N. Prezerakos, N.D. Tselikas, and I.S. Venieris, "Context-aware service engineering: A survey," *Journal of Systems and Software*, vol. 82, Aug. 2009, pp. 1285-1297
- [20] R. Robinson, K. Henriksen and J. Indulska, "XCMML: A Runtime Representation for the Context Modelling Language", Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops, 2007. PerCom Workshops '07, pp. 20 – 26, 2007
- [21] P. Zimmer, "A Calculus for Context Awareness," BRICS Report Series 2005
- [22] L. Birkedal, M. Bundgaard, T.C. Damgaard, S. Debois, E. Elsborg, A.J. Glenstrup, T. Hildebrandt, R. Milner, and H. Niss. "Biographical programming languages for pervasive computing," In *International Workshop on Combining Theory and Systems Building in Pervasive Computing*. Position Paper, May 2006
- [23] International Standard ISO/IEC 15909, "High-level Petri Nets - Concepts, Definitions and Graphical Notation", 2000
- [24] "Knowing What's What and What's Not: The Five W's (and 1 "H") of Cyberspace". (<http://www.media->

1 awareness.ca/english/resources/special\_initiatives/wa\_resources/wa\_shared/tipsheets/5Ws\_of\_c  
2 yberspace.cfm) Media Awareness Network. Accessed September 12, 2008.

- 3 [25] I.A. Moschakis and H.D. Karatza, "Towards scheduling for Internet-of-Things applications on  
4 Clouds: A Simulated Annealing approach", *Concurrency and Computation: Practice and  
5 Experience*, 2013
- 6 [26] Five W's, ([http://en.wikipedia.org/wiki/Five\\_Ws](http://en.wikipedia.org/wiki/Five_Ws)), Accessed: 23<sup>rd</sup> January 2008
- 7 [27] M. Bazire and P. Brézillon, "Understanding context before using it," *Modeling and Using  
8 Context*, 2005, p. 29–40
- 9 [28] S.L. Kaini, A. Anjum, M. Knappmeyer, N. Bessis and N. Antonopoulos "Federated Broker  
10 System for Pervasive Context Provisioning", *Journal of Systems and Software*, 2013, 86(4), pp:  
11 1107-1123
- 12 [29] M. Fouquet and G. Carle "AmbiSense - Ambiente Vernetzung Mobiler Sensoren", Flyer  
13 published by University of Tübingen, March 2006
- 14 [30] K.E. Kjær. A Survey of Context-aware Middleware. In: 25th conference on IASTED  
15 International Multi-Conference: Software Engineering 2007, pp.148-155
- 16 [31] S.Jang, W.Woo, "5W1H: Unified User-Centric Context ," The 7th International Conference on  
17 Ubiquitous Computing, 2005
- 18 [32] X. H. Wang, D. Q. Zhang, T. Gu, and H. K. Pung, "Ontology Based Context Modeling and  
19 Reasoning using OWL 3 . CONON: The Context Ontology," in *Proceedings of the Second  
20 IEEE Annual Conference on Pervasive Computing and Communications Workshops*, 2004, pp.  
21 18-22
- 22 [33] J.W. Kaltz, J. Ziegler, and S. Lohmann, "Context-aware Web engineering: Modelling and  
23 applications," *Revue d'intelligence artificielle*, vol. 19, 2005, pp. 439–458
- 24 [34] A.K. Dey and G.D. Abowd, "Towards a better understanding of context and context-  
25 awareness," *CHI 2000 workshop on the what, who, where, when, and how of context-  
26 awareness*, Citeseer, 2000, pp. 1–6
- 27 [35] A. Zimmermann, "Context Management and Personalisation: A Tool Suite for Context- and  
28 User-Aware Computing", PhD Thesis, 2007
- 29 [36] J. Pascoe, N. Ryan, and D. Morse, "Human-Computer-Giraffe Interaction – HCI in the field",  
30 *Proceedings of the Workshop on Human Computer Interaction with Mobile Devices*, Glasgow,  
31 Scotland, 1998
- 32 [37] A. Schmidt, K. Aidoo, A. Takaluoma, U. Tuomela, K. Van Laerhoven, and W. Van de Velde,  
33 "Advanced Interaction in Context", 1th International Symposium on Handheld and Ubiquitous  
34 Computing (HUC99), *Lecture notes in computer science*, vol. 1707, Springer, pp. 89-101, 1999
- 35 [38] P.J. Brown, J.D. Bovey, and X. Chen, "Context-aware applications: from the laboratory to the  
36 marketplace," *Personal Communications, IEEE*, vol. 4, 2002, pp. 58–64
- 37 [39] N. Ryan, J. Pascoe, and D. Morse, "Enhanced reality fieldwork: the context-aware  
38 archaeological assistant", In Gaffney, V. et al. (Eds.) *Computer Applications in Archaeology*,  
39 1997
- 40 [40] H. Hettema and T. A. F. Kuipers, "The periodic table — its formalization, status, and relation to  
41 atomic theory," *Erkenntnis*, vol. 28, no. 3, pp. 387-408, 1988
- 42 [41] S. Garzonis, "Usability Evaluation of context-aware mobile systems: A review," 3rd UK-  
43 UbiNet Workshop, 2005
- 44 [42] T. Gross and R. Klemke, "Context modelling for information retrieval—requirements and  
45 approaches", 2003.
- 46 [43] F. Wang, E. Camacho, & K. Xu, "Positive influence dominating set in online social networks"  
47 in *Combinatorial Optimization and Applications*, 412(3), 313–321.  
48 doi:10.1016/j.tcs.2009.10.001, 2009.
- 49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65