

# Defining and Representing Activity Context for Systems Analysis

**Dan Diaper**

DDD SYSTEMS, Bournemouth, BH11 8SZ, U.K.  
ddiaper@ntlworld.com

## Abstract

Representing context information associated with people and digital devices performing activities is presented using a formal systems model based on a legal but simplified version of set theory. A five set Venn diagram, the PentaVenn diagram, allows analysts to work using a graphical logic rather than with equations. Model symmetry is shown to facilitate identifying different types of context, tangible and intangible.

## Introduction

This short paper is one of a pair that attempts to fulfill the brief from the 2011 AAAI Activity Context Representation Workshop organizers to provide “an authoritative introduction to the topic”; the topic being this paper’s title and its other half’s, ‘A Formal Systems Approach to Machine Capture, Representation and Use of Activity Context’. Limitations of page space precludes a comprehensive literature review, so rather than an inadequate one, none is offered at all. The second paper uses the concepts introduced in this one in a modest worked example that addresses context capture by digital devices.

Elaborating on this paper’s title it is necessary to establish who or what is going to define context, and what sort of things are going to be represented, i.e. a definition of context. The ‘who’ are assumed to be software professionals involved with the design of digital devices who use a structured design method. This paper’s aim is to add context to what is already modeled.

Currently and generally, such systems models contain things, such as people, digital devices and other objects associated with activity performance, and relationships between the things in a systems model. Things can be tangible, physical objects or they may be intangible, most usually representing information processing in people’s minds or in machines’ software. Relationships are either structural or communicative. Structural relationships define

how things are related, e.g. A is\_a\_part\_of B, or C is\_separate\_from D. Communicative relationships represent transfer between things, most usually modeled as information (see Diaper, 2004).

To address what is going to represent context, then, for the workshop’s purposes, the assumption is that digital devices will represent context that they will use appropriately. Such use may be automatic or in collaboration with one or more users. The concept of capturing context thus has two major, distinct meanings referring to: (i) systems modelers as described above; and (ii) digital devices.

Inverting the problem of defining what is context, ‘What makes context different from what is already modeled?’ This paper’s answer is that context is like what is already modeled, except that it affects the work performing parts of a system but is not itself affected by these other system parts. In this view, when context is created or modified, this is done by a different system from when context operates on the main system of interest.

## General Systems Modeling

As computers became mainstream business systems in the 1970s and onwards, the early evidence was of unreliable systems that failed to meet the new owners’ and users’ expectations. Exception handling in particular was poor to nonexistent and just its cost could easily exceed any financial benefits of the new technology. The basic solution was structured design methods that provided representational formats and methods of working with these. The emphasis was on understanding the requirements, converting these into a design which could be faithfully programmed, and the resulting device could then be evaluated with respect to its programming (bugs), design and to its requirements.

A change over these decades is what is modeled as the system of interest. Initially it was the computer system with a user’s terminal hung off one side of the model, then there was a slow swing to user-centered design. Taken too literally, the latter is imbalanced. While modeling the human elements of any system is vital, the general goal is to

design and build devices, so these must be represented as being of equal importance to a system's human parts.

During World War II, the Rand Corporation developed a general systems approach. Checkland's Soft Systems Methodology (SSM) is a widely known derivative, which can provide systems models, from multiple perspectives, of many things in their environments. While the early work had a sound mathematical basis, SSM is soft on mathematics/logic and soft in its use, hence describing itself as a methodology rather than a method. SSM therefore takes considerable, expert skill to do well and whether it is done well is hard, or impossible, to tell, i.e. in this it provides poor cognitive support to systems analysts.

Diaper (2000) has shown that SSM representations can be expressed formally without losing SSM's desiderata. He uses set theory, and a legal subset of it called Simplified Set Theory (SST), for a number of reasons: (a) set theory is the basis of all modern mathematics; (b) Venn diagrams provide a graphical logic as an alternative to set theory's algebraic manipulations; (c) similarly, SST can be used to formally reason about a system without recourse to such algebra; and (d) for the mathematically challenged, SST can be converted into English like sentences by the choice of suitable set names and some simple symbol-to-lexical substitutions. A new addition to this list is a solution to how to discuss systems modeling independently of any particular structured design method and its representational forms and logical formalisms. Set theory, or SST, is the solution proposed because it is so mathematically fundamental. The idea being that any formal systems modeling approach can be expressed set theoretically, although a more advanced logic, e.g. a first order predicate calculus, will need other logical additions.

### Set Theoretic Systems Modeling

In the broad, interdisciplinary field that is Human Computer Interaction (HCI), a common systems model is the user-computer interface dialogue model (Figure 1). Using Dowell and Long's formulation (e.g. 1989), in this model the user is a work system (WS) and the computer the application domain (AD). The WS's user performs work by changing the AD and work is successfully achieved if the AD is changed to meet the user's goals. Communication between WS and AD is via the user interface (UI), which can be separated into input and output interfaces.

A set theory equivalent to Figure 1 is shown in Figure 2 as a Venn diagram. The two sets are the UWS and the CAD and their intersection  $\{UWS \cap CAD\}$  is the UI. The communicative relationship arrows of Figure 1 have no place in set theory as arrows, but can be represented as elements within the UI intersection and, for now, can be left as text descriptions, e.g. 'user keystrokes are converted

to ASCII'; the arrows are included in Figure 2 only to illustrate its equivalence to Figure 1.

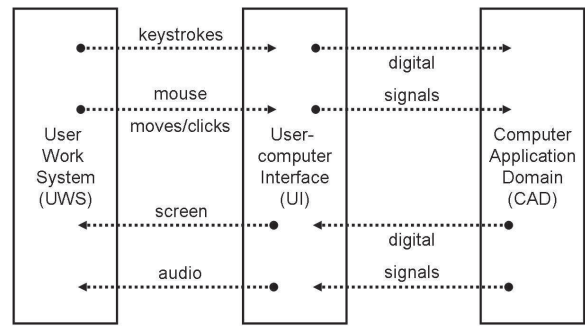


Figure 1. Traditional user-computer interface dialogue model.

If one were only in possession of the Figure 1 model and wanted to add context to the model, then there are lots of different options. The simplest would be just to put the whole of Figure 1 inside a big box marked context (the basic SSM solution, for example), but a different alternative would be to have three context boxes and these could be drawn either inside or outside the existing boxes in Figure 1. To redraw Figure 1 to include context thus depends on systems analysts making theoretical decisions. They are provided with little help from their existing systems model/diagram; it is not clear what theoretical decisions need to be made, never mind exploring the possible consequences of each, alone and in combination.

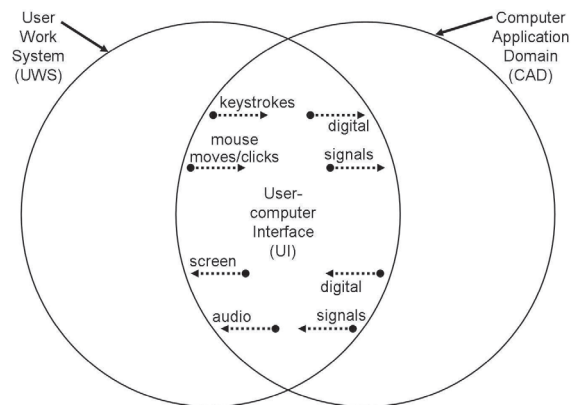


Figure 2. Venn diagram user-computer interface dialogue model.

In contrast, with a Venn diagram it is easy to decide that there is only one possible, sensible way to represent context, and that is: (i) as another set,  $C = \{\text{Context}\}$ , as shown in Figure 3; and (ii) that this new set intersects with the  $A = \{UWS\}$  and  $B = \{CAD\}$  sets. Alternative models are all logically incomplete as the maximum number of intersections (including where a set intersects with no other, e.g. areas 1, 2 and 3 in Figure 3), is  $2^N - 1$ , where  $N$  is the number of sets. The Figure 3 Venn Diagram has seven such 'intersections', each of which can be expressed logically by a different set theory equation. All the alternative mod-

els have fewer, but not logically different, intersections, i.e. all alternative models are only versions of the Figure 3 one in which some of the seven intersections, by definition, are empty ( $\emptyset$ ).

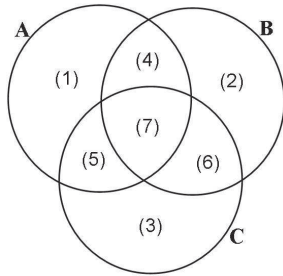


Figure 3. Three set Venn diagram.

Figure 3 only establishes a general theoretical solution to the representation of context. Neither Venn Diagrams or set theory algebra has been much used for systems analysis and there are problems with both. The number of sets that can be drawn for a legal Venn diagram is extremely limited, rarely greater than five.

Whatever the divers problems of the mathematically challenged for whom SST was invented, there is a subtle problem with writing set theory equations as one often has to attend to specifying parts of the whole system that are not currently relevant. In Figure 3, for example, if interested in area (1), then this is everything that is in set A that is not in sets B and C, i.e.  $A \setminus (B \cup C)$ . With many more than three sets, then this negative focus of specifying what is not currently of interest is often tedious and inappropriate. To appreciate the problem, a systems model of twelve sets, about the size often used by the author, has associated with it a maximum of 4,095 different equations ( $2^{12}-1$ ). SST overcomes this negative focusing as a consequence of its simplification of set theory. SST directly identifies intersections and, because it eschews algebraic reduction, every fully expressed SST equation has  $2^N-1$  parts, each part representing an intersection. SST's equations are only practical because it uses the symbol  $\setminus^*$  to mean: *and excluding all the intersections not specified to the left*. Thus, area (1) in Figure 3 becomes  $A \setminus^*$ , which if its  $\setminus^*$  were expanded in SST would contain the union of the other six intersection equations, i.e. the union of Figure 3's SST equations (2) to (7) below, but with all their  $\setminus^*$ s collapsed to a single one.

- (1)  $A \setminus^*$       (2)  $B \setminus^*$       (3)  $C \setminus^*$
- (4)  $\{A \cap B\} \setminus^*$     (5)  $\{A \cap C\} \setminus^*$     (6)  $\{B \cap C\} \setminus^*$
- (7)  $\{A \cap B \cap C\} \setminus^*$

In SST everything is either a set or an element of a set and its main simplification is that all sets intersect in all possible ways. Thus, set elements correspond to terminal leaf nodes in a tree representation and a major part of using an SST systems model is locating set elements correctly in their unique intersection. Venn diagrams could help this enormously, but three sets isn't enough for practical systems modeling. Diaper (2000) produced a Venn diagram,

the Fruit Bowl Model (center-bottom, Figure 4), of regular shapes where five sets intersect in every possible way, including not intersecting with any other set.

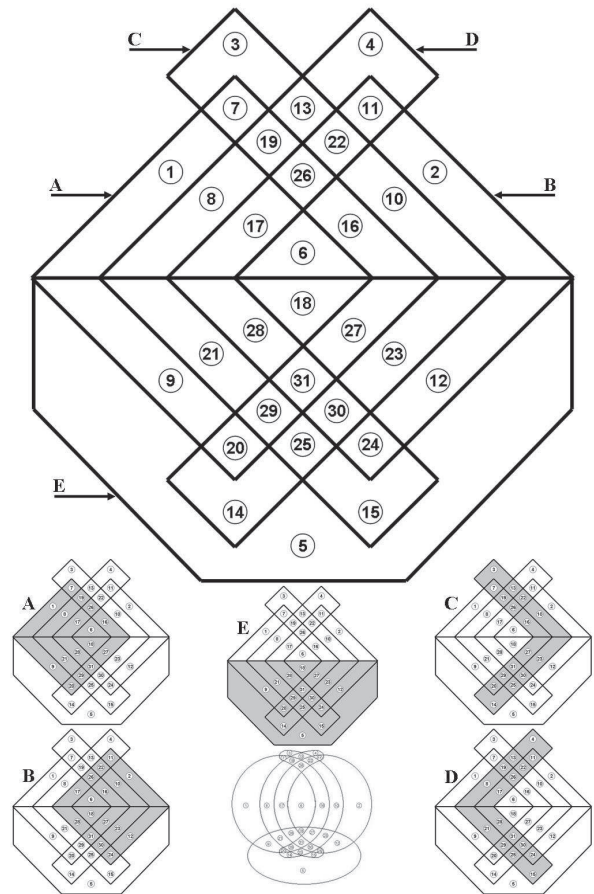


Figure 4. PentaVenn diagram where each of the 5 sets intersects in every possible way (31 numbered areas). The 5 sets are illustrated below. The original Fruit Bowl Model is bottom-center.

The author carries a sheet of colored Fruit Bowl Models in his wallet because, he claims, they let him graphically solve logic problems. The limitation is still five sets, but generally this seems sufficient for examining relevant parts of larger systems models. However, the Fruit Bowl Model as a piece of graphic design is rubbish and is difficult to use on paper, if easier online where intersection areas can simply be colored in during analysis. It has been redrawn, and renamed the PentaVenn diagram, so to make the intersection areas a more useful size, while still using regular shapes. Figure 4's PentaVenn diagram shows all the 31 intersections numbered (following the Figure 3 convention), and at the bottom the shapes of the five sets.

Returning to the user-computer interface dialogue model, the PentaVenn diagram model extends that of Figure 3 by separating the user interface into User Input (to the computer) and Computer Output (to the user), and representing each with their own set. A sensible assignment of the sets is the left diaper (a synonym of diamond) as the work system (A) and the right diaper as the application

domain (B). The right pointing angle as the work system output to the application domain (C) and vice versa the left pointing one the from the application domain to the work system (D). The six-sided lower set then represents context (E).

In this use of the PentaVenn diagram, intersection elements can represent one-way communicative relationships. Consequently, to break an activity down into a sequence of steps, the two set intersections provide paths through the model, as illustrated in the second paper.

Furthermore, for identifying relevant context, only five sets are necessary as context can influence everything else in the model, but can't itself be changed because if it were, it would be part of the work system, the application domain, or of their two interfaces, e.g. the weather may be a context for laundry drying activities, but hanging washing out to dry does not affect the weather. While less physical contexts, e.g. beliefs, have to be created and be modifiable, the argument is that in such cases the context change is the effect of a work system on an application domain representing the context, and which can then only subsequently operate as context, i.e. in a one way fashion on the main system of interest's work performing parts.

Finally, in the Dowell and Long formulation of systems models, (1989, see also Diaper and Sanger, 2006), work systems and application domains will usually contain more than just a user, computer and their interfaces. The PentaVenn diagram and its SST equations provides a generally useful means of selecting levels of description.

## Modeling Activities

A newer term for 'tasks' and 'task analysis' is 'activities'. It is moot how substantial is this change, but tellingly, the primary, first representation of virtually all task analysis methods has been referred to since the 1960s as an 'activity list' (see Diaper, 2004). An activity list is a structured, prose description of an activity. It differs from a scenario (see Diaper, 2002a; 2002b) by being a list of short sentences that describe an activity's steps, e.g. (1) Put on underwear. (2) Put on trousers.

This getting dressed type of example is fine for a single instance of an activity because things happen in time. For more than one instance, then it is necessary to deal with alternative sequences, options, interruptions and simultaneous, parallel activities, e.g. watching television while getting dressed, (see Diaper and Stanton, 2004).

No one denies that people do the same thing in different ways, yet this universal view holds a nightmare of complexity for systems analysts. Even the two steps getting dressed example can illustrate many difficulties. First, the sequence has no options, but Superman is one person who is an exception case (underwear outside). In many activities there are alternative step sequences, e.g. milk before or

after pouring the tea, and optional steps, e.g. sugar or not (which may be quantified by lumps or spoons). Is underwear optional? Obviously it is, for some people always, and for some people in some circumstances. So now our activity list might read:

(1) [OPTION] **Put on underwear;**

[EXCEPTION: Superman];

[EXCEPTION: Never wear underwear].

(2) **Put on trousers.**

The first optional step means that the sometimes underwearless person must make a complex cognitive decision that will involve context, past, present and predicted future, as indicated by the "in some circumstances".

There are also class and levels of description issues with this example, e.g. what does "trousers" cover as a category? Are shorts included with trousers, and how long in the leg before shorts become trousers? Furthermore, using the term 'trousers' shows gender and cultural biases, e.g. ignoring women's skirts and Scotsmen's kilts.

Analyst bias probably is ubiquitous and arguably unavoidable, e.g. in cross-cultural cases. Cognition and context are problematical for those with a software development background. Diaper (2002a; 2002b) has proposed his All Thought Is Scenario-based (ATIS) hypothesis as a basis for reasonable fidelity mental models suitable for use by non-psychologists. ATIS proposes that when thinking, what we actually do is tell ourselves stories, i.e. scenarios, rather than thinking logically. We might imagine different people getting dressed, and so imagine female or Scotsman scenarios, and we tend to place our scenarios in settings, i.e. in an activity's immediate physical context and sometimes in psycho-social or other intangible contexts. For example, when first thinking of the watching television while getting dressed example above, the author imagined a young male in a small lounge and a sofa, but the author would not know the color of the sofa unless asked, which reflects how we build useful mental scenarios without imagining every detail. The ATIS hypothesis argues that generating and evaluating scenarios is a powerful way to think, and it is how people think, but it comes at a cost of relatively frequent, and in some circumstances predictable, errors of omission and commission.

The systems modeling approach described in the previous section helps systems analysts structure their thinking and encourages analysts to generate a wider coverage of scenarios than might otherwise be produced. This is done by what the author calls 'populating the systems model' and is illustrated in the next section.

## Populating a Systems Model

Imagine a software engineering project scenario where a device will help people to remember to put on their under-

wear appropriately. This is not so foolish, if it is part of a much larger digital home care system for those with genetic or acquired cognitive difficulties, e.g. Downs syndrome, dementia.

Even a cursory examination should convince analysts that just the getting\_dressed\_system is a very complex one. Also, it is dealing with non-standard people and some of these need expert human care, for example balancing self sufficiency against meeting acceptable social norms. Starting from scratch, the general HCI approach would be to go and talk to and, if possible, watch what the carers do. Emphasizing the carers leads to one systems model where they are the work system,  $A=\{\text{Carer}\}$ , and the cognitively challenged people (the cared-for) are the application domain,  $B=\{\text{Cared4}\}$ ; one of the work system's goals is to ensure that the cared-for are acceptably dressed (N.B. 'acceptably' is a complex concept). Using a PentaVenn diagram, the angled sets become: (i) what the carer might do,  $C=\{\text{Act}\}$ , and the cared-for perceive; and (ii) what the carer perceives of the cared-for,  $D=\{\text{See}\}$ , and any behaviors directed by the cared-for at the carer. The bottom, six sided set remains as the activity's context,  $E=\{\text{Context}\}$ .

The model can now start to be populated by generating a scenario and working through the model, for example, starting by considering the carer and where 'Professional' is a note to the systems analyst to identify professional knowledge and skills and 'Social knowledge' a note, for example, about acceptable dress (e.g. underwear inside):

**(1) Carer**\*

'Professional'  $\in$  (1)

'Social knowledge'  $\in$  (1)

Psychologically, the carer will prepare or switch to looking at their person professionally (8) and then making their observations (17):

**(8) {Carer  $\cap$  See}**\*

'Prepare for expert observation'  $\in$  (8)

**(17) {Carer  $\cap$  Cared4  $\cap$  See}**\*

'Observe cared for person's state'  $\in$  (17)

We could now add another element to (1) as the result of such an observation, e.g. 'Assessment knowledge'  $\in$  (1). Obviously we also need to model the cared-for person:

**(2) Cared4**\*

'Cared for person physical'  $\in$  (2)

'Cared for person mental'  $\in$  (2)

'Clothes'  $\in$  (2)

'Getting dressed'  $\in$  (2)

What makes this systems approach powerful is that early on analysts need to consider the intersections that are empty, i.e. those that they have not populated. For example, considering area (11), a psychologically orientated analyst might consider the actual state of how the person is dressed as opposed to how the carer perceives it:

**(11) {Cared4  $\cap$  See}**\*

'Actual visible state of cared for person'  $\in$  (11)

Then the analyst can add an element to the cared for person (2) about their beliefs about how they are dressed. Most of the fifteen areas in the top half of Figure 4 can be populated and the heuristic should be to be reluctant to prematurely mark intersections as empty.

There is nothing novel about the above, except for providing more support than just starting with just a couple of boxes for the carer and cared-for and a few arrows between them. Many of the things modeled are intangible, to do with knowledge, skills, perception, plans and goals, and alternative definitions of work system and application domain can systematically divide these (deduction) or combine sets for induction, so producing different levels of analysis. SST equations can be written for systems models with many sets and the PentaVenn diagram provides a focus on five of a model's sets and all their possible relationships, including those that are impossible ( $\emptyset$ ), i.e. no scenario can be observed, described or imagined that can populate an empty intersection.

The lower half of the PentaVenn diagram represents activity context and its symmetry means that for every populated intersection above the line, their should be corresponding context below the line. Thus for the first few intersections listed above, (1), (8), (17) and (2), they have examples of context such as the following few, of many possible ones:

**(9) {Carer  $\cap$  Context}**\*

'Professional qualifications'  $\in$  (9)

'Professional experience'  $\in$  (9)

**(21) {Carer  $\cap$  See  $\cap$  Context }**\*

'the carer knows how to be "in the right frame of mind" to see one of their people'  $\in$  (21)

**(28) {Carer  $\cap$  Cared4  $\cap$  See  $\cap$  Context }**\*

'There is sufficient light to see'  $\in$  (28)

**(12) {Cared4  $\cap$  Context}**\*

'Cared for person is in their usual room'  $\in$  (12)

'Cared for person may become emotional'  $\in$  (12)

'Suitable clothes are usually available'  $\in$  (12)

'Usually the person can dress themselves'  $\in$  (12)

Rarely do such elements emerge directly from their source data and how a structured systems analysis approach can provide cognitive support for analysts can be illustrated with the last element above, 'Usually the cared-for person can dress themselves'  $\in$  (12). This led the author to consider when both carer and cared-for share a context that is not part of their perceptions or actions, which would be area (18):

**(18) {Carer  $\cap$  Cared4  $\cap$  Context}**\*

'The carer and cared-for share the belief the latter can usually dress themselves'  $\in$  (18)

Then, since this belief is shared, then each must believe it, which would add similar elements to (9) and (12). Furthermore, on the basis of the PentaVenn diagram's top-bottom symmetry, (18), (9) and (12) should now have ad-

ditional elements in (6), (1) and (2), respectively. This illustrates a problem with analyzing context, that it is present in an activity but is often invisible to analysts until there is a performance failure or major change to draw it to analysts' attention. It is hard enough with physical context, e.g. 'Who includes the office floor, before there is an earthquake?', but identifying psycho-social contexts is even harder, although for engineering purposes the generative theories, e.g. human memories are recreated as required and not searched for as in a computer, are mostly not relevant. It is even possible for analysts to completely ignore whole areas of context, such as economic ones in the example above, and this is particularly so if over reliance is placed on observational data rather than imagined scenarios, e.g. complete system failure could be caused by paid carers being made redundant, which isn't likely to be observed.

The above analysis is not yet complete, even for the initial scenario. Further scenarios, imagined or based on interview/observational data, will further populate the systems model. When to stop analysis is usually determined by project schedules and resources, rather than the ideal that new scenarios don't change the model significantly. Furthermore, given the aim is to design a digital device to wholly or in part replace the carer, then a systems model of the device needs to be developed. If the carer is to be replaced, then a device set can be swapped in for the carer set. If the device is to work with the carer and cared-for person, then a seven set model is needed, i.e. adding Device and its Interface. Writing SST equations for seven sets, or quite a few more, is no harder than for five sets and, in any case, intersection equations that are specified by a large numbers of sets tend to be empty, i.e. they cannot be populated. This is the author's experience over a decade and is supported by the hypothesis that where many things interact, then their effects are diluted, ultimately to the point of no effective effect.

## Conclusion

As engineering, systems analysis is not science, but an enterprise that needs to satisfy project criteria. A systems model needs to be fit for purpose, but, as in other types of engineering, different analysts may produce different systems models that meet the same criteria.

In summary, activity context is different from those parts of the system involved with achieving work, because context can affect such other parts of the system but cannot itself be changed by the work related parts.

Activity context can be represented in the same way as other parts of a systems model, as in done in this paper. Systems models that don't explicitly represent context already represent many types of thing, tangible and intangi-

ble. The PentaVenn diagram's vertical symmetry means that context will also represent a similar range of types. Indeed, most context types will be intangible, relating to people's individual and shared beliefs, knowledge and skills, and devices will need to capture, represent and emulate these as well as dealing with physical contexts. Much context is invisible and the problem of identifying all of it is insoluble, but analysts can achieve better coverage if supported by structured methods.

There has not been space to discuss the advantages of having formal systems models. Since set theory is at the foundations of mathematics and logic, then other formal systems modeling approaches can adapt the approach described as they will be using set theory, howsoever it might be disguised in more advanced notations. Also, SST and the PentaVenn diagram can be used with non-formal systems analysis methods, e.g. as Diaper (2000) did with SSM.

This paper has not included discussion of the English like expression of SST equations designed to help the mathematically challenged because they are unlikely to appreciate the paper's formal content.

The second paper for the AAAI Workshop provides a more thorough, simpler worked example and addresses how digital devices can capture, represent and use activity context information. Its conclusion, in line with this first paper's, is, 'Context? Do Not Panic.' because, mostly, context is like what is already modeled.

## References

- Diaper, D. (2000) Hardening Soft Systems Methodology. in McDonald, S., Waern, Y. and Cockton, G. (Eds.) *People and Computers XIV*. 183-204. Springer.
- Diaper, D. (2002a) Scenarios and Task Analysis. *Interacting with Computers*, 14, 4, 379-395.
- [Diaper, D. (2002b) Task Scenarios and Thought. *Interacting with Computers*, 14, 5, 629-638.
- Diaper, D. (2004) Understanding Task Analysis for Human-Computer Interaction. in Diaper, D. and Stanton, N.A. (Eds.) *The Handbook of Task Analysis for Human-Computer Interaction*. 5-47. Lawrence Erlbaum Associates.
- Diaper, D. and Stanton, N.A. (2004) Wishing on a sTAr: The Future of Task Analysis for Human-Computer Interaction. in Diaper, D. and Stanton, N.A. (Eds.) *The Handbook of Task Analysis for Human-Computer Interaction*. 603-619. Lawrence Erlbaum Associates.
- Diaper, D. and Sanger, C. (2006) Tasks For and Task In Human-Computer Interaction. *Interacting with Computers*, 18, 1, 117-138.
- Dowell, J. and Long, J. (1989) Towards a Conception for an Engineering Discipline of Human Factors. *Ergonomics*, 32, 11, 1513-1535.