# Co-evolving problems and solutions

## The case of novice interaction designers in Botswana and the UK

Nicole Lotz*, Helen Sharp*, Mark Woodroffe*, Richard Blyth*, Dino Rajah**, Turugare Ranganai **

** The Open University, UK, nicole.lotz, helen.sharp, mark.woodroffe, richard.blyth@open.ac.uk*
*** Botho University, Botswana, dino.rajah, turugare.ranganai@bothocollege.ac.bw,*

Abstract: This paper establishes that problem-solution co-evolution is observed in novice interaction designers in the UK and Botswana. However, in the majority of Botswana protocols we could see a new type of co-evolution, which we termed solution-problem co-evolution. Solution-problem co-evolution uses 'off the shelf'' solutions to structure the problem space. Both types of co-evolution are described and discussed in this paper. The findings are drawn from the analysis of 18 (5 UK, 13 Botswana) 1-hour design protocols from two cohorts of students studying the same undergraduate Open University Interaction Design module, one in Botswana and one in the UK. Participants were required to complete a medical interaction design task under controlled conditions. We based our analysis on a coding scheme that was developed specifically for this protocol study. The coding scheme is based on Schön's seminal work on reflective practice. It visually represents activities in the problem and solutions spaces.

***Key words: Design thinking, Problem-solution co-evolution, Solution-problem co-evolution, Novice designers, Schön***

## 1. Background

For some, design is regarded to be a problem solving activity [16], while others term design a solution-focused activity [7, pp 103]. As simple as it sounds, the objective of designers is to search for solutions to solve the problem. But how do designers do that?

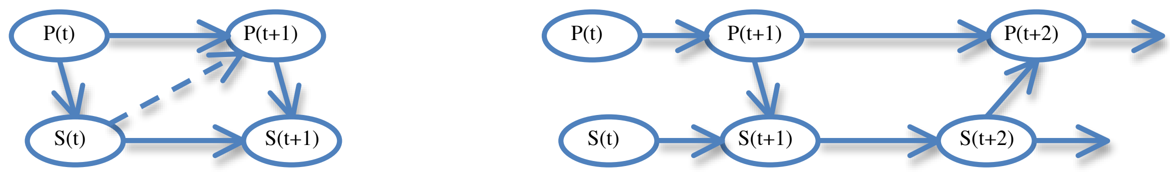



Figure 1 (a) Redrawn model by Maher and Poons [12] and (b) redrawn model by Dorst and Cross [5]. In both models, Problems (P) and Solutions (S) evolve over time (t+1, t+2). Arrows from right to left indicate evolution, arrows up or down indicate shifts from problem to solution space or vice versa.

Maher and Poon [12] argued that because design is dealing with wicked problems, solutions couldn't be found by a simple nor straightforward search. In order to solve difficult problems, intermediate solutions 'talk back' to the designer to help understand and define the problem. This reformulation of the problem leads to solution proposals being advanced. So Maher proposed a model of design exploration as coevolution of problem and solutions (Figure 1a). Empirical engineering design research around the same time [11] confirmed that solution conjectures are used to evolve (explore and understand) the problem. They also found that intermediate solutions like prototypes give experimental feedback and new constraints to help redefine the solution. A few years later based on protocols studies of expert designers, Dorst and Cross [5] refined Maher's model and termed this phenomenon problem-solution co-evolution (Figure 1b). Problem-solution co-evolution describes the "iteration of analysis, synthesis and evaluation processes between the two notional design 'spaces' – problem space and solution space." (pp 434).

A few years back, empirical work around this topic was picked up by Smulders, Reyman and Dorst [17], who used the DTRS7 protocol data of architectural design and identified that in design team discussion, usage (use scenarios) mediated between problem and solution trajectories. They suggested working towards a typology of problem-solution co-evolution.

We developed a method for coding interaction design protocol data, based on Valkenburg and Dorst's [18] analysis of reflective practice in design teams. Valkenburg and Dorst's as well as our method to code protocol data is based on Schön's seminal work and well-known theory of reflection in action [14]. Schön views every design task as unique. Designers must frame a design situation to impose coherence. The view of the world created by the designer is constructed. The results of the designer's actions are evaluated and the current view of the design task interpreted. Schön describes this cycle of action and evaluation as a 'reflective conversation with the situation'. Designers name factors in the problem space and solution space, frame a problem by creating a context for viewing it in, take actions to move towards a solution and evaluate those actions. Schön describes this as Naming, Framing, Moving and Reflecting. The actions described by Schön are very similar to what Cross and Dorst [5] describe as the iteration between analysis, synthesis and evaluation in the co-evolution processes.

Our paper contributes to the effort to characterize different types of problem-solution co-evolution. Previous studies looked at expert designers and design teams and teams of mixed expertise in western countries – the US and UK. Findings in this paper are based on protocol studies with novice designers in the UK and in Botswana.

## 2. Our study

The use of protocol data to analyse the cognitive aspects of design behaviour has a long tradition in design research and has revealed great insights into 'design thinking'. Traditionally, design thinking is defined as a cognitive ability pertaining to the process of designing [1]. Describing the ability of expert designers to synergise ideas by evolving the problem and solution space in parallel has been a major advancement in design research [5]. The application of problem and solution co-evolution has not been reported in novice designers.

### 2.1 Setting

For several years, The Open University has been partnering with Botho College, Botswana to deliver a degree programme of Computing and its Practice. This programme includes the module 'Fundamentals of Interaction Design' which is delivered in the UK and Botswana using exactly the same teaching materials. For the OU, this means a core text book [15] plus four books of additional exercises and teaching material; the module is designed to be studied by distance education students on their own, and hence they are both prescriptive and self-contained.

15 protocol studies of novice interaction design behaviour were recorded in Botswana in January 2012 and 6 in the UK between July and September 2012. Two protocols in Botswana and one in the UK could not be used for analysis. In Botswana, thirty participants were selected from a pool of volunteer students taking the Interaction Design course at Botho College. In the UK, volunteers studying the same module were self-selected following a call for participation in this constrained design task.

Participants in Botswana were paired, while UK participants worked individually. Participants practiced talking aloud (and working together in the case of Botswana) before they were given the design problem, which was concerned with how to support sick people at home in administering the right medication at the right time. The participants were given an hour to design solution(s) or partial solution(s) to the medication design problem working closely with each other and talking out loud about what they were doing and thinking. The study was conducted in the participants' chosen language (Setswana or Kalanga) and English in the UK and a facilitator was present to prompt participants to talk aloud if necessary. Botswana protocols were translated into English for analysis.

## 2.2 Coding method

We will briefly introduce our developed coding method to demonstrate how we identified the types of co-evolution in our data. Valkenburg and Dorst's [18] analysis method interested us because it seemed to be a relatively generic descriptive method that we hoped could be applied to the domain of interaction design. The visual notation produced during the coding process offers an overview of the vital elements of a design project and reveals patterns of behavior.

The research team is composed of researchers from different design and computing domains as well as psychology. Initially, each of four coders coded two translated protocols from Botswana using Valkenburg and Dorst's description. There was overlap such that each coder had both of their protocols also coded separately by two of the other coders. The coding led to poor agreement between the coders over what constituted Framing, Naming and Moving, although Reflecting seemed more readily identified. In the initial coding, Analogy was also identified as a major source of our coder variability. We had different interpretations of the code definitions. After several iterations of coding and discussing we then agreed on following definitions:

- Framing: providing a focus for the discussion. Frames generate new design requirements or maybe give a set of constraints.
- Naming: identifying specific things you have to design or design for, such as functions and components.
- Moving: exploring the problem space; expanding the brief; adding detail.
- Reflecting: evaluating a design idea, concept or requirement. Is this a good thing to be doing? Why is it good?
- Analogy: transferring elements from the familiar to use it in constructing a novel idea.

The code category for Analogy was taken from Christensen and Schunn's [3] studies of the use of analogy in engineering design. They differentiate between problem, explanatory and solution analogies.

In order to code more consistently, our analysis was guided by the concept of frame signature matrix. A signature matrix clarifies the narrative structure of a frame [8]. Such narratives are diagnostic as well as prescriptive stories. Thus frames guide both analysis and synthesis in practical situations. In any particular frame-narrative, Gamson and Lasch [8] identify two elements: the 'framing devices' suggest how to think about an issue; the 'reasoning devices' suggest what should be done about the issue. The framing devices correspond to what we call problem space and reasoning devices to what we term solution space.

A frame's distinctive signature is revealed by studying the beliefs and meanings of the designers (as revealed through their 'designing' verbal protocols), and interpreting them in terms of their professional practice and their culture. Repeatedly reading through our protocols, we identified elements of the frame structure which constituted both framing and reasoning devices along core constructs of focus from the domain of interaction design. Framing devices were: notions of the centrality of the user; the importance of user goals and characteristics; the nature of the environment of use; and the nature of the user tasks. Elements of the frame matrix, which constituted a reasoning device, are for example methods to understand the possible situation and solutions (e.g. storyboarding), solution components (e.g a screen, a power off button), and conceptual frameworks for identifying solutions (e.g analogy). A more in-depth description of the development of a part of the signature matrix enhanced coding method is given by Blyth et al [2]. Using this signature matrix-supported notation we achieved a higher coder agreement. Any further disagreement was resolved in discussion with the researcher team.

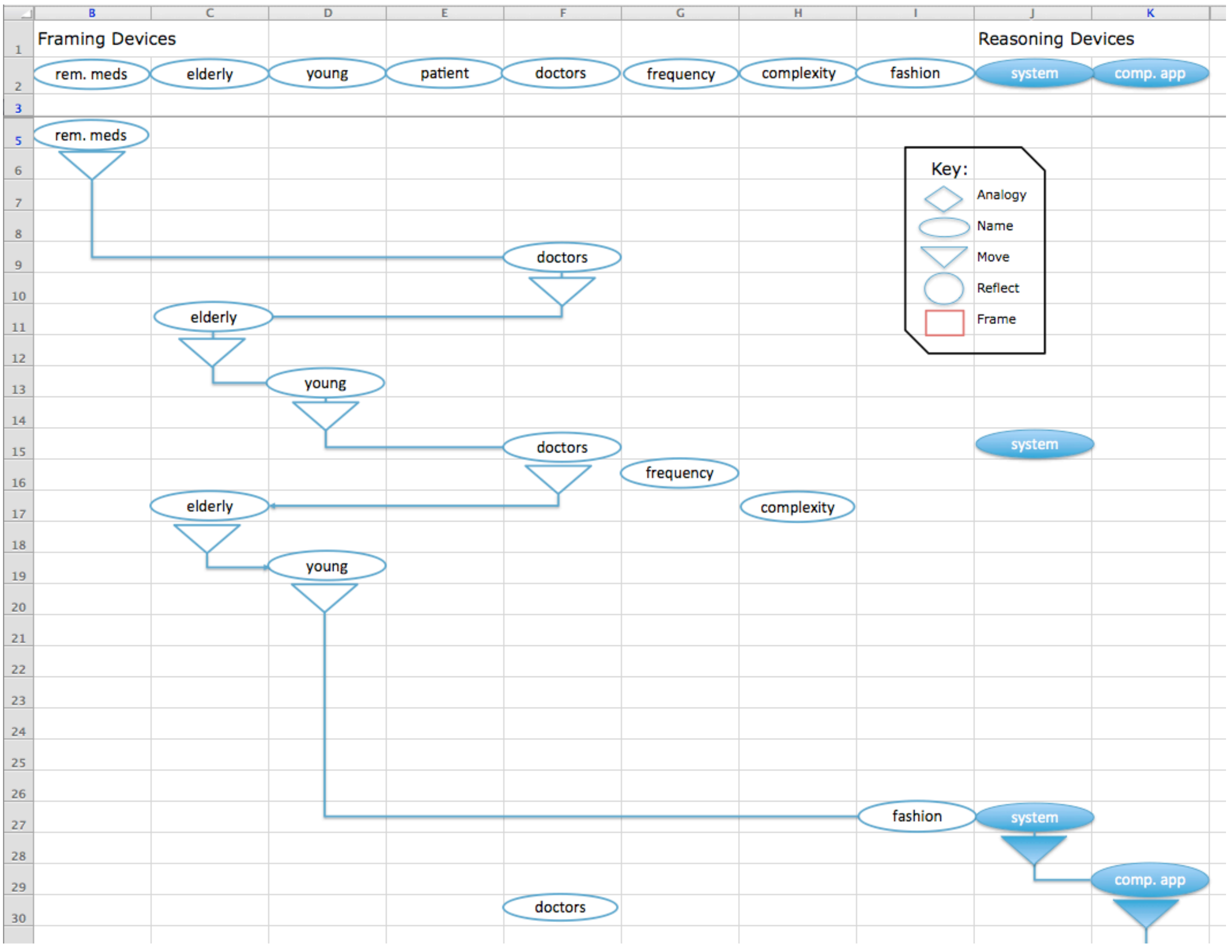



Figure 2. Coded extract of UK Participant 7 beginning of the protocol (rows 4-30). Outline shapes signify framing devices and filled shapes are reasoning devices. Column A contains the transcript, which is hidden in this excerpt for the benefit of readability of the symbols.

To capture these devices we noted each instance of Naming, Moving or Reflecting and the use of Analogy in a unique cell in a spreadsheet for each protocol. We were using the same symbols as Valkenburg and Dorst did: an oval for Naming; a triangle for Moving; a circle for Reflecting. We added a diamond for the use of Analogy. A Frame was captured as a bounding box. We used different colours to differentiate frames. Figure 2 shows an edited (condensed) version of the coding of the first utterances in UK participant 7 (UKP7) protocol. The header (rows 1-4) shows the accumulation of all devices used in the protocol.

In Figure 2, row 5-10, UK Participant 7 (UKP7) starts with: "*Yes. Okay. We have a problem with forgetting to* *take medication, and what I need to do is produce a solution for this problem. What I think for the users, they can just divide the users into three groups. Generally three groups of users… Someone has to set up the solution of what it will be in order to devise a system to remind people to take drugs. So I think the doctors will be responsible for that. The second group is elderly people. Most of them are taking drugs and so a solution for them. The third group of people that is mentioned in the problem description are young people."* In column B row 5 the design goal 'reminding medication' is named and hence coded with an outlined oval. Then also the focus moves to user groups in F9 'doctor', C11 'elderly' and D12 'young' and back to doctor. All these are framing devices; they belong to the problem domain. Each time the participant moves forward after naming a new device, the triangle is used. We have connected the devices with a solid line indicating the flow of the thought process. Although in row 15 the participant names the reasoning device 'system' (filled oval shape) the focus remains on the users and in the problem space.

Frequency (H16) and complexity (I17) of drug taking are named and the focus shifts from the doctor to the elderly users in C17 before it settles on young users for a wile (D19-29). The transcript for rows 19-29 is: "*Younger people, it is not the same as older people, they have a better memory. That is my assumption because we can meet both kinds of people with* ___[not understandable]. *But they do many more things during the day time and can be forgetful as well. Another thing is the part of the system, part of the assumption which is for younger people must be fashionable. It must be something that they really want to use."* You can see in the transcript that 'fashionable' is captured in J27 and system in K27. The last 2 sentences mark a shift from a focus on framing devices to reasoning devices.

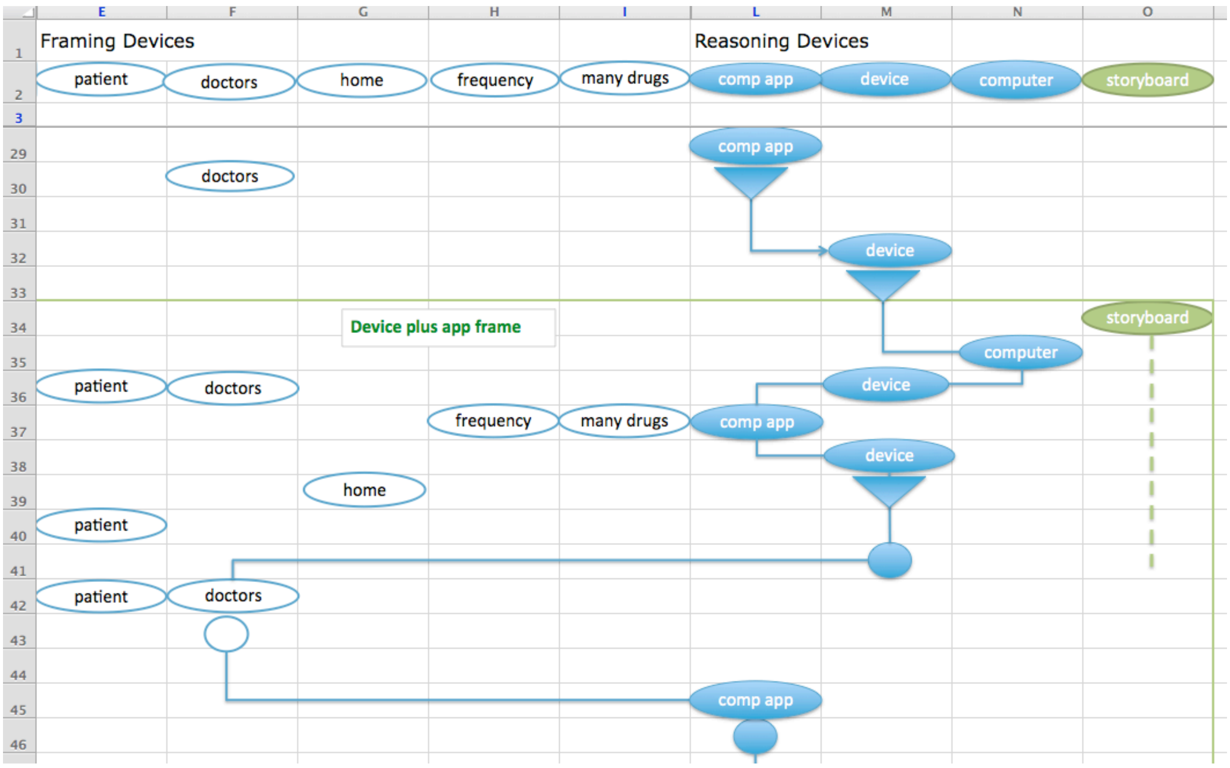



Figure 3. Coded extract rows 29-43 of UK Participant 7. Continued from Figure 2. Some framing devices are hidden in this excerpt for clarity of illustration.

Figure 3 shows that in row 29 UKP7 names the next reasoning device 'computer application' (L29): *"So,* *going from this point of view where you can make the assumption that actually there must be two ends of the system at least to function of the solution. The first part is computer based, or just simply an application, which generally would be used by doctors to allow them to set up drug taking times. The second part, which I don't know now, I can say it is about technology but possibly a bit different technology than the computer one. It is a bit of equipment or device, anything like that, which will remind patients about the times of taking drugs."* For now UKP7 remains in the solution space.

In Figure 3 in cell M32 a 'device' is named and coded. In row 33 we have indicated the beginning of a frame 'device plus app' with a green bounding box (note the box is very large and therefore only shown partially). The participant is evolving this frame using a storyboard: "*Alright. Generally, I think that if I have to draw the storyboard it will look more or less as that. The patient is coming to the doctor. The doctor is sitting at the computer and prescribing some drugs. After that, the doctor with the use of our product, is setting up times for taking drugs and for each drug. After that, the patient receives the device.* The parallel activities of talking and drawing are indicated with a solid line for the spoken and dashed line for the drawing activity. Naming of the reasoning device 'storyboard' starts the dashed line from O34.

From row 41, UKP7 looks back at the storyboard: *"Okay. The story that I have, actually I can raise a few situations which in various forms to be assigned. The first situation of the patient coming to the doctor is actually redundant for our design, because it doesn't involve any application or device. It is just simply the patient coming in. Prescribing drugs is a part which is, again, redundant for our design because I am not meant to design that bit of the system."* With this, the participant moves from the solution space back to the problem space naming doctor (F42) and patient (E42) again and reflects about (circle symbol) what problems need to be addressed or can be disregarded. UKP7 continues: "*It is my assumption that there is already a computer based system where the doctor can input the patient's data and simply put in the drug's name and it will print the prescription or whatever. So I guess this system has already been done by someone else and it exists."* With this, the focus is moved back to the solution space, being indicated by the solid line moving across in row 45.

In the header (rows 1-4), all framing and reasoning devices are added as they occur, and can be reused if they are recurrently named in the transcript. Progressing with this coding through all 19 protocols allows us to clearly mark activity, the focus of activity and the shift of focus between problem and solutions spaces in each protocol. The visual notation includes the identification of a flow of discussion, denoted by a solid line that allows us to focus on interesting episodes and see patterns of behavior. Especially for the study of co-evolution, it reveals the parallel activity in problem and solutions spaces and also shows the shift of focus between these spaces in the verbalization.

## 3. Findings

### 3.1 Problem-Solution co-evolution

We can see from the Cross and Dorst's model (Figure 1(b)) that there are two aspects to co-evolution:

1. A continuous and parallel evolution of problem and solution spaces.
2. A shift of focus from problem to solution space and vice versa. The first shift originates in the problem space.

Dorst and Cross [5] called a shift in focus between problem and solutions spaces 'building bridges'. Using our analysis method, this shift can be seen very clearly following the solid line. UKP7, which was used to explain our coding procedure and notation in the previous section (Figures 2 and 3), reveals a shift in focus (the solid line is drawn) from the problem to the solution space in line 29 where a bridge is built between users and specific requirements and a system, which is then specified as an application and device. This is the second aspect of problem-solution co-evolution.

In the following rows the first aspect of problem-solution coevolution, the parallel evolution of problem and solution spaces, can be observed. In Figure 3 rows 34-40, UKP7 draws a storyboard where a computer application is designed for a doctor, and the device for the patient is mentioned too. The focus of activity is in the solutions space. But we see a parallel naming of framing and reasoning devices in these rows, which shows a co-evolution phase.

From row 42, a bridge from the solution space back to the problem space is built in which UKP7 reflects on what was outlined in the storyboard. The participant refines the problem by reflecting on the interactions between doctor and patient. The conclusion for the design brings us back into the solution space in row 45. UKP7 reveals both aspects of problem-solution co-evolution in the first 50 lines of transcript.

What we can also see in this excerpt is what Cross and Dorst called evaluation, and Smulders et al [17] refined as the mediator 'use'. The discussion of 'use' (user behavior) during storyboard drawing helps co-evolution. The participant moves in the solution space, but also mentions problem criteria in parallel. The subsequent reflection of the user behavior allows the building of bridges from solution space to problem space and back.

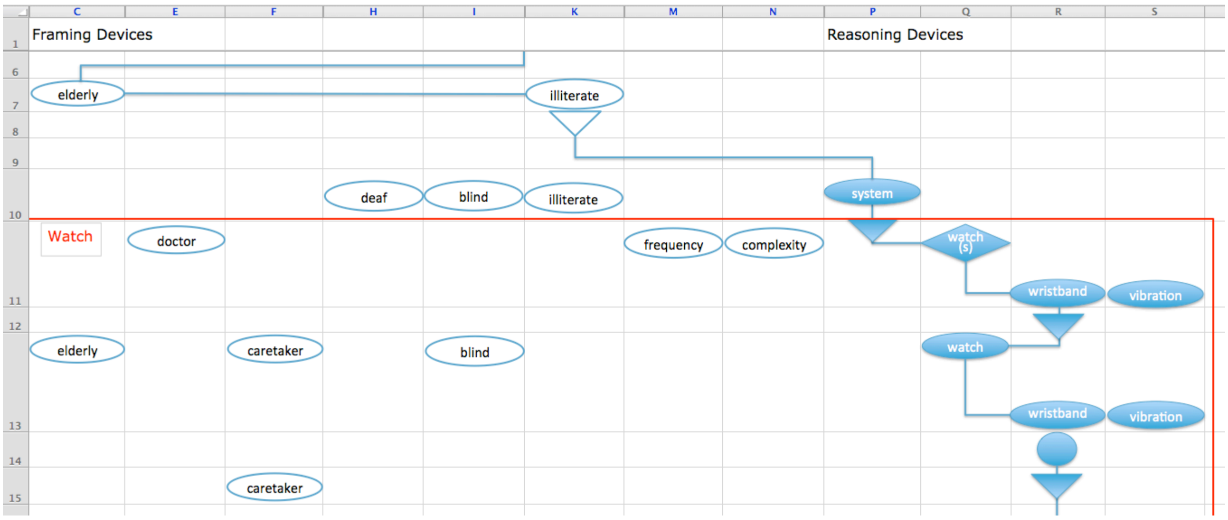



Figure 4. Botswana pair 7 edited excerpt, short after beginning of the protocol (rows 6-15). Some framing devices are hidden in this excerpt for clarity of illustration.

We could also observe problem-solution co-evolution in some Botswana novice designers. In Figure 4 is shown that Botswana pair 7 (BP7) starts their exploration in the problem space similarly to UKP7. They name several framing devices in the beginning. Figure 4 shows 'elderly' (C7) and 'illiterate users' (K7). From illiterate the first bridge is built. BP7A says: *"Let's make a system that, e.g. since we have different kinds of patients and some of them might be deaf and some of them blind, hence our product has to be able to be used by even a blind or deaf patient and those who can't read should be able to use it"* (row 10). BP7B replies*: "Yeah. You have right idea, and we can design something sort of a watch and can be worn on a wrist, which is already programmed by* *the doctors. A doctor can prescribe medication which has different medication timings e.g. five different kinds of medications and a patient wearing the wrist-band product preprogrammed will be reminded about the prescription timings by the product vibrating"* (row 11). BP7B introduces the solution 'wristband product' (R11) supported by a solution analogy 'sort of watch' (diamond in Q11) and adds a solution component 'vibration' (S11).

Although now being focused on the solution exploration, the designers add more framing devices to the problem space in parallel during and after crossing the bridge. For example, 'deaf' (H10) and 'blind' (I10), reiterating 'illiterate' (K10), mentioning the role of the 'doctor' (E11), 'frequency' (M11) and 'complexity' of medicine taking (N11), 'elderly' users (C13), 'caretakers' (F13), and so on. There is a wealth of detail developed in the problem space in parallel to the solution space, on which the focus is set primarily. Again we can see that the discussion of use of the proposed products is a mediator between the two in parallel evolving spaces.

### 3.2 Solution-problem co-evolution

We have identified a new type of co-evolution, which we term solution-problem co-evolution. Co-evolution can be generally defined as refining together the formulation of a problem and solution ideas. Solution-problem co-evolution highlights different aspects:

1. A continuous and parallel evolution of solution and problem spaces.
2. A shift of focus from solution to problem space and vice versa. The first shift originates in the solution space.

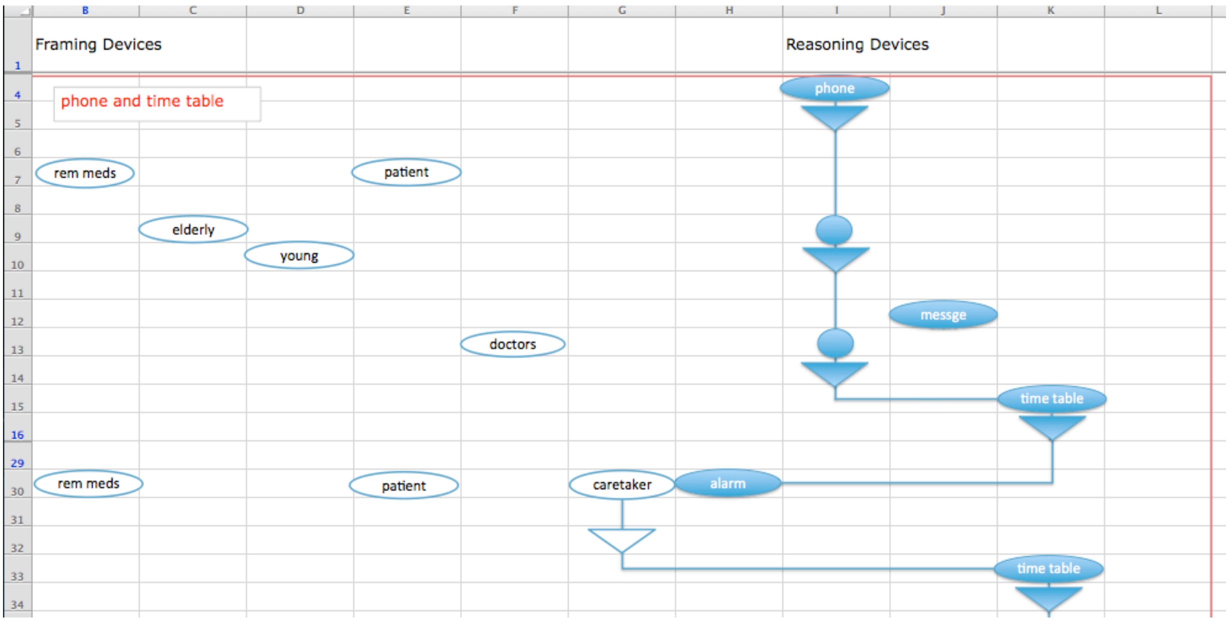



Figure 5. Botswana pair 12 edited excerpt of the coding of the beginning of the protocol (rows 4-34)

The difference might sound subtle, but it has substantial effects on ideation. Several Botswana designers suggested a solution first before they looked at the problem criteria. Lawson's seminal work on architecture students revealed a solution focus in design based problem solving [10]. Literature reports varying approaches in novices. Christiaans and Dorst [4] found that some novices became focused on information gathering rather then solution finding, while others did not collect many information at all and solved simple problems. However, it is unclear from literature if those students immediately suggested solutions and used the solution to structure the problem space. This is what we have found in Botswana novices.

To illustrate solution-problem co-evolution, Figure 5 shows an edited excerpt from the beginning of Botswana pair 12 protocol (rows 2-34). BP12A suggests in rows 4-7: *“Nowadays, Nowadays everyone has a phone, isn’t it? Including the patients and everybody else so we can use a phone to remind them to take their medication.”* Here the designer names phone as a solution to the problem of reminding the patient to take medication. But note, the problem criteria are not analyzed before the solution is suggested, and both spaces evolve in parallel. BP12B reflects on this idea in row 9: *”In what manner because not everyone knows how to use a phone as a reminder for their medication, isn’t it that medication is also taken by elderly people.”* The solution gives structure to the problem space, but the designer pair remains with their focus (solid line linking symbols) in the solutions space. BP12A counters in row 12: “*What I mean is there can be a way in a phone that can remind them eg. Message alert or….*” They now name a new solution component ‘message alert’ (J12). BP12B in row 13: *“Yeah, I understand if you talk about message alert where will the message come from, will it come from doctors or the patient has to create the message alert?”* BP12A in row 14: *“The message alert can be created by the patient or it can come from the doctors knowing that they have to send message to a particular patient at a specified period of time reminding the patient to take their medication”*. Even here, problem and solution structuring still occur in parallel. Solution and problem co-evolve without having analyzed the problem or structured the problem space beforehand.

BP12B comes up with a new idea in row 15: *“Yeah, if it’s like that its clear, Again, I was thinking patients and doctors can come up with a timetable for each patient’s medication. Each patient can then be given a time table so that each time the patient takes the medication they can indicate by marking the timetable that they took their medication.”* There is some discussion on the timetable idea.

In rows 30 and 32 a weak bridge to the problem space is built by BP12B: *“They can set the alarm. You see, again I think that caretakers should be instructed to always remind the patients to take medication on time, you see. And to always go for medical checkups on time. So I don’t know but we can decide what to choose, which one you think we should choose?*” They agree to choose timetable and merge the idea with a phone later on. Arguably one could say no bridge has been built at all. In fact very few solution components are named thereafter. Co-evolution halts until the designers reflect on the use of the timetable-based design solution.

Figure 6 shows how the designers *BP12B* reflect on the use of timetable from row 185: *“Patient indicates by marking the chart, to show that he/she took his/her medication. A way which was set by the doctor. Ok no, this was set by the patient isn’t it. A way which can alert patient on their medication, they mean that patient should know the importance of taking their medication accordingly. Isn’t it? Isn’t it that our duty is to come up with a way which can be used to alert patient and reminding them about their medications? You think that thing.”* BP12A: *“What do they mean, does it mean that patients are told how the medicine is used ….is used for.”* BP12B: *“No, the dangers, I think here it means our duty is to come up with a way that can be used to alert and remind patient about their medication. It’s like it’s similar to training patients on the importance of taking their medication. You see…*” They name a new design goal ’training people’ (B191) using a problem analogy, and then start suggesting solutions to the redefined problem. BP12B: *“In hospital and doctors should select a day, isn’t it…”* BP12A: *“Patients should be given pamphlets to read during their spare time.”* (I202)

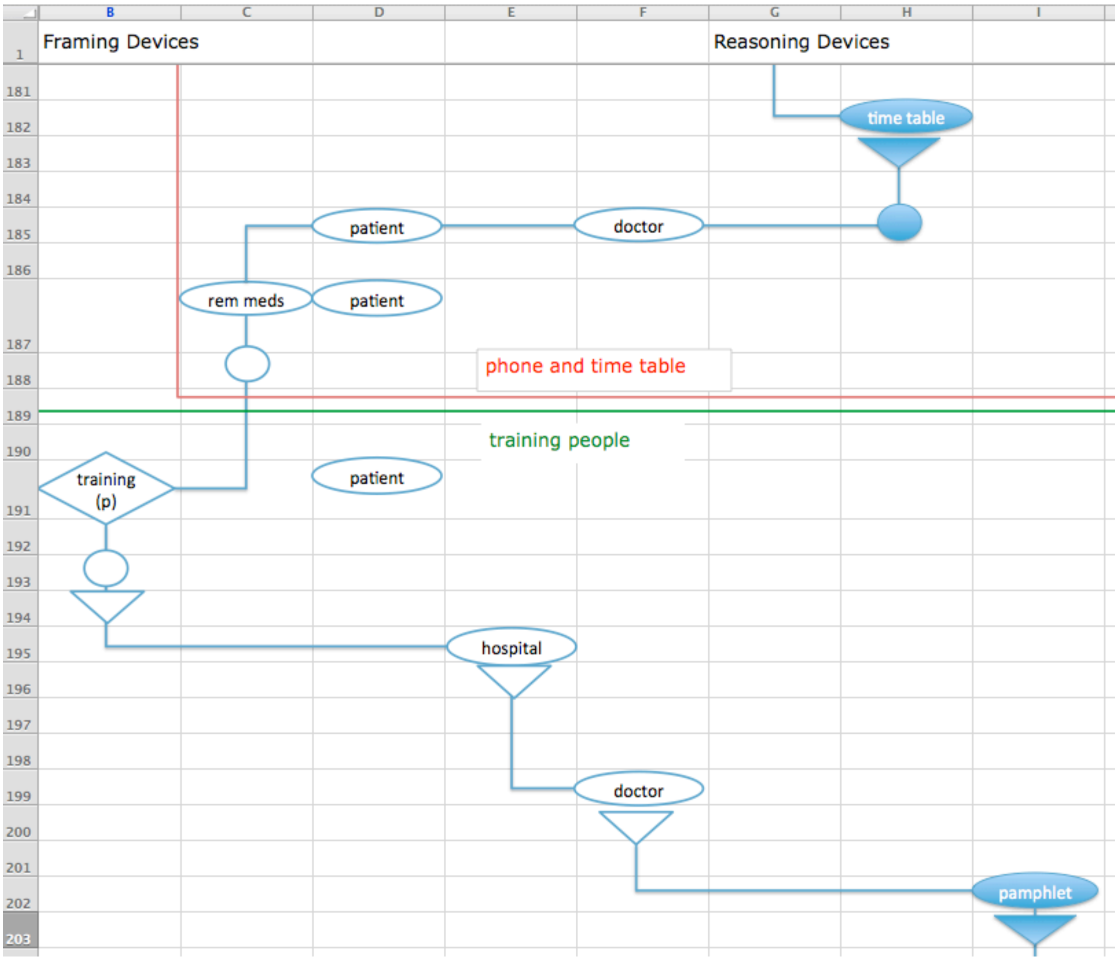
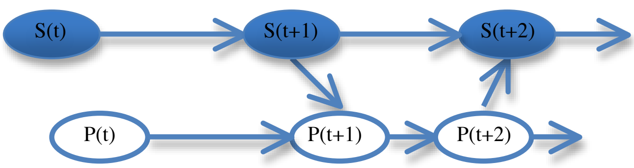



Figure 6. Botswana pair 12 after reflecting on the timetable based design, builds a bridge between solution and problem spaces leading to the radical reframing of the design problem and goal (rows 181 – 203)

Designer BP12B was inclined from the beginning (row 30) to shift the problem. During subsequent reflection on the use of the solutions ‘timetable’ and ‘phone’, the designers build a bridge to the problem space by radically reframing the design goal from ‘reminding medication’ to ‘training patients’, and started naming radically new solutions like ‘Pamphlets’ and later ‘Drama’ and ‘Songs’ that are broadcast on the radio and TV etc.




S(t) initial solution space

S(t+1), S(t+2) (re)structured solution space

P(t) initial problem space

P(t+1) P(t+2) (re)structured solution space

Figure 7. Solution-problem co-evolution model

The majority of the 13 Botswana protocols start with the solution. The solutions are everyday objects and services that are around them. They use simple ‘off the shelf’ solutions to structure the problem space. Figure 7 shows this schematically. A solution is proposed first, before some problem criteria are named. This constitutes the initial solution and problem spaces S(t) and P(t). A bridge from the solution to the problem space is built S(t+1) – P(t+1), where the problem is reframed and a bridge is built back to the solution space P(t+2) – S(t+2).

Another example to illustrate this model is Botswana Pair 4. The designers start with a bracelet idea, which along with a watch takes them to a smartphone and finally to a generally defined 'handheld device'. The problem space evolves in parallel considering solutions for the young and busy workers – the phone, which is not ideal for elders. They then build a bridge to the problem space in which they discuss the complexity of the problem, i.e. frequencies of drug taking, different environments and for different users. The problem was slightly restructured but not radically reframed as we have seen in Botswana pair 12, because the design goal didn't change. In further co-evolution the pair created a hybrid solution taking components from all named ideas, and adding detail through reflection on user behaviour, creating a watch-like, handheld object with screen and audio capture for playback as the reminder.

## 4. Conclusion

We could observe Dorst and Cross' [5] co-evolution model in our data of novice interaction designers in the UK and Botswana. Novice designers use problem-solution co-evolution. The co-evolution and linking of problem and solution spaces is supported by reflection on user behaviour. Dorst and Cross termed this kind of reflection 'evaluation' and Smulders at al [17] further defined it as the mediator 'use'. Reflection captures more than evaluation of user behavior (i.e. reflection on self or on circumstances). Further work could explore more objects and uses of reflection, i.e. alongside Hong and Choi's [9] model of the dimensions of reflection, to further define the mediator 'use' in problem-solution co-evolution.

Our signature matrix enhanced coding method allowed us to visually represent where co-evolution phases and shifts from problem and solution spaces occur. This led us to discover a new type of co-evolution, which we have termed solution-problem co-evolution. This type of co-evolution could only be observed in novice interaction designers in Botswana. In solution-problem co-evolution a simple solution from everyday life of the designer is used as a departure point to structure the problem space in parallel to the evolving solution.

This observation could help us to study design thinking in emerging markets where previously the concept of 'innovation from the bottom/base of the pyramid' has been described in business literature, i.e by Prahalad [13] Innovation is observed in the modification of everyday objects to solve emerging problems with a shortage of resources. More work should be done to explore the design thinking process behind such strategies.

## Acknowledgements

This work is funded by the Leverhulme Trust. We would like to thank all our participants.

## 5. References

[1] Badke-Schaub, P.G., Roozenburg, N.F.M. and Cardoso, C. (2010) *Design thinking: a paradigm on its way from dilution to meaninglessness?* In Dorst, K., Stewart, S., Staudinger, I., Paton, B. and Dong, A. (Eds.), Proceedings of DTRS8, pp 39-49.

[2] Blyth, R., Schadewitz, N., Sharp, H., Woodroffe, M., Rajah, D., & Turugare, R. (2012) *A frame signature matrix for analysing and comparing interaction design behavior* [Online PDF], In Proceedings of BCS HCI12, Available at <http://ewic.bcs.org/content/ConWebDoc/47817> [Accessed March, 2013]

[3] Christensen, B. and Schunn, C. (2007) *The relationship of analogical distance to analogical function and preinventive structure: The case of engineering design*, vol. 35, Springer, New York.

[4] Christiaans, H. and Dorst, K. (1992) *Cognitive models in industrial design engineering: A protocol study*, In Taylor, D.L. and Stauffer, D.A. (Eds), Design Theory and Methodology - DTMS92, American Society of Mechanical Engineers, New York.

[5] Dorst, K. and Cross, N. (2001) *Creativity in the design process: co-evolution of problem–solution*, Design Studies, vol. 22, no. 5, pp 425-437.

[6] Cross, N. (2004) *Expertise in design: an overview,* Design Studies, vol. 25, no. 5, pp 427–441.

[7] Cross, N. (2007) *Designerly ways of knowing*, New York, Birkheuser

[8] Gamson,W, and Lasch, K. (1983) *Evaluating the welfare state: social and political perspectives,* In Shimon, S. and Yuchtman-Yaar, E. (Eds.), Evaluating the welfare state: social and political perspectives. Academic Press, New York.

[9] Hong, Y. and Choi, I. (2011) *Three dimensions of reflective thinking in solving design problems: A conceptual model,* Educational Technology Research and Development, vol. 5, pp 687-710.

[10] Lawson, B.R. (1979). *Cognitive strategies in architectural design*. Ergonomics, vol. 22 no. 1, pp 59-68.

[11] Kolodner, J.L. and Wills, L. M. (1996) *Powers of observation in creative design*, Design Studies, vol. 17 no. 4, pp 385-416.

[12] Maher, M.L. and Poon, J. (1994) *Modelling design exploration as co-evolution*, Microcomputers in Civil Engineering on Evolutionary Systems in Design, vol. 11, 195-210.

[13] Prahalad, C.K. (2012) *Bottom of the pyramid as a source of breakthrough innovations,* Journal of product innovation management, vol. 29, no. 1, pp 6-12.

[14] Schön, D.A. (1983) *The reflective practitioner*, Basic Books, New York.

[15] Sharp, H., Rogers, Y. and Preece, J. (2007) *Interaction design: Beyond HCI*, 2nd Ed, John Wiley, London.

[16] Simon, H. (1996) *The science of the artificial*, (3rd Ed), MIT Press, London.

[17] Smulders, F., Reyman, I. and Dorst, K. (2009) *Modeling co-evolution in design practice,* In Proceedings of ICED 2009, (pp 335-346), Stanford University. Stanford, CA, USA.

[18] Valkenburg,R. and Dorst, K. (1998) *The reflective practice of design teams,* Design Studies, vol. 19 no. 3, pp 249-271.