Creating and Evaluating Problem Templates for Problem Generation Within the Context of Stroke Cognitive Rehabilitation

COSC 366 Final Report

Scott Ogden[1]
Supervisor: Prof Antonija Mitrović[2]
Sjo75@uclive.ac.nz[1] | tanja@cosc.canterbury.ac.nz[2]
Computer Science & Software Engineering
University of Canterbury
10 February 2012

Abstract

Stroke Rehabilitation would be more effective if the patients conducted activities personalised to them, as opposed to a set of generic activities which may be irrelevant. This project has the intention of creating problem templates, which will be able to describe the conditions that must be met for a stroke patient to complete an activity, for a wide range of personalised activities. The project is part of the larger Stroke Rehabilitation System, which the ICTG lab is working on. This customisation of treatment is the motivation for the project.

The concept of problem templates is used extensively in this project. Problem templates are "*chunks of domain-specific knowledge, compiled mentally by experts, and used to solve commonly occurring problems in a particular domain*" [1]. Because of their dynamic nature, it is possible to create very generalised problem templates, which can be applied to create very specific scenarios. This research aims to create many of these templates, with the intent of showing that it is possible to create a varied set of potential problem scenarios, using only a few problem templates for each task.

After completing background research, a system for developing these problem templates was developed, and six such problem templates created. The six problem templates were for everyday tasks that stroke patients might perform, such as 'make hot drink', 'make frozen meal', and 'make sandwich'. A quick survey rendered a number of specific examples to instantiate the problem templates, such as: 'make coffee', 'make pizza' and 'make tuna salad sandwich'. From preliminary examination of these problem scenarios, it would appear that these problem templates can be used to generate problem scenarios, in well-defined situations. For the majority of the specific examples, the resultant dependency graphs of states were the correct solution, or required only minor changes. The Author suggests that further problem templates should be made and that further studies be conducted so as to maximise its effectiveness for the Stroke Rehabilitation Project in the long term.

Contents

Table of figures

Acknowledgements

The Author would like to thank Moffat Mathews, for not only giving him this opportunity, but also for all the guidance, and helping him through the entire process. Thanks to Prof Antonija Mitrović for her supervision, proof reading, and encouragement. Thank you to Louise Ogden for her last minute proof reading. Finally thanks to Jay Holland for assessing the problem scenarios, and for all your feedback.

1 Introduction

1.1 The importance of stroke rehabilitation

Stroke is the third largest killer in the United Kingdom, and the single largest cause of severe disability [2] . Of those that do survive for more than one year; 43% were handicapped [3]. Even though much of the brain may be damaged after a stroke, the brain is remarkable, in that it can create connections through other functions to compensate for the loss. Indeed this is the focus of most optimal rehabilitation strategies [4]. It is recognized that there is a definite relationship between the intensity of the rehabilitation process, and the benefits reaped for the patient [5]. is particularly evident in the first year when most of the function is recovered.

1.2 Motivation for the Research

For the stroke rehabilitation process to be as effective as possible, it would be best if the tasks could be tailored to the patient's individual tastes and life style [6]. Stroke patients often have very limited focus and energy [7], therefore, with a personalized process, they could spend more of their rehabilitation focusing on other areas, which are likely to benefit them in real life. The process of creating a fully customized routine with every single required state modified 'by hand' would be an incredibly time-consuming one, even by a skilled operator, with each seemingly simple task requiring dozens of constraints. Problem templates are one potential method of overcoming this barrier.

This research evaluates the usefulness of problem templates [1] in problem generation. While techniques similar to problem templates have been used for problem generation before, this is generally in the context of digital systems [8], not stroke rehabilitation, or anything medical for that matter. With only marginal effort, problem templates should allow for the generation of problems unique to the patient. This is because it is possible to insert any arbitrary variable into a general template to make it unique for the patient. The main goal for the research, is to assess whether this can be done consistently.

1.3 An Overview of the Research

Everyone's routine is unique; everyone does slightly different activities at slightly different times. This may lead to the question of what is the *right* way for someone to conduct their routine. In fact there is no one *right* way to conduct their routine. One cannot simply say that drinking Milo at 8am is wrong; you must instead drink coffee at 8.05am. Instead, what this research aims to develop is problem templates which will, to continue this analogy, allow the user to make any instant hot drink at any time. However, it must also provide feedback to the user that will help guide them in whatever it is that they choose to do. If they begin to make coffee, but they forget to turn on the jug to heat the water, this might fail the task requirements, and they may receive feedback on this topic. The approach adopted for giving this feedback and modelling the user's behaviour in the larger stroke rehabilitation project is Constraint Based Modelling (CBM) [9], and is slightly outside of the scope of this research. Instead, this research focuses on creating 'problem templates' for the problems. These problem templates consist of states; which are a mere step away from constraints, i.e. the underlying knowledge-representation formalism used in CBM. Problem Templates are general enough that they are a template for solving any one of a range of similar problems. The states can then be specialised by applying them to a specific action. This can generate a dependency graph of states, applicable to the task at hand, and useful for giving feedback.

When this system is combined with the Stroke Rehabilitation System outlined in Section 2.2, it will allow the larger system to generate dynamic scenarios in real time, which will be specific for the patient's lifestyle.
We hypothesise that this can in fact be achieved; that it is possible to develop these problem templates for the context of stroke rehabilitation.
This was assessed by an expert in the problem domain; full details about the research are available later in the report.

1.4 The Structure of the Report

Section 2 of this report regards background information, and the greater context of the system. While it cannot cover all the information from the surrounding domains for obvious reasons, it does intend to include all information pertinent to comprehending this report in context.

The goals and hypothesis of this research are outlined in Section 3. This also regards the design and implementation of the research study, along with a number of problem templates, rendered as dependency graphs. The description and discussion of the evaluation of the study is given in Section 4, while the conclusion and future direction can be found in Section 5. This is followed by a bibliography and appendices.

2 2 Background and Related Work

2.1 Strokes

A stroke, or 'brain attack' [10], is the result of an obstruction of a major artery in the brain, or a leakage of blood in the brain. It results in the death of brain cells in the affected tissue [11]. It can affect a patient mentally, physically and emotionally. Stroke patients can often be affected with difficulty speaking or communicating [12], difficulty with movement [13], and/or general cognitive difficulty [14]. This can range from drooping of the eyelid, through to patients unable to move or communicate. For the purpose of the project, it is assumed that the stroke patients involved in the programme will have the use of both arms and legs, and a reasonable level of brain function.

2.2 Stroke Rehabilitation Project

The context of this research is for it to work with the greater Stroke Rehabilitation Project, being undertaken in the ICTG lab. This project aims to use a virtual reality environment, to help patients practise everyday tasks and build their prospective memory. The virtual reality environment will mimic the patient's home, and the tasks are to be similar to everyday life. There have been similar studies in the past, such as [15], which simulated a virtual shopping trip, but these have been focused on testing the patient's memory, not improving it.

2.2.1 Prospective Memory

Prospective memory is used to remember the need to complete future actions [16]. It is often described as the need to "remember to remember" [17]. For this to happen, there needs to be not only perspective memory, but also conscious thought, and so this is severely impaired in patients with memory or cognitive disorders [16]. In the past, there have been many trials to determine the exact impact of various types of brain trauma, including strokes. Examples of these studies include [18], [19], [20], [15], and [6]. It is a general conclusion that strokes generally impair prospective memory, and not purely from the decrease in brain function.

Prospective memory is usually broken up into two components: time- and event-based prospective memory [17]. However, in other papers, event based has been split into event and activity based prospective memory, such as in [18]. Time-based memory is for activities such as taking pills at a set time each day, or turning the oven off exactly one hour after it is turned on. Of event types, there are two distinct sections [17]: immediate execute, and delayed execute. Immediate execute events are, as the name would suggest, performed immediately. These are tasks such as returning a book to its owner when you see the owner. These however, are rare in real life. What is observed in real life is a delayed execution, as the person finishes what they are doing, and only then begins another task [21]. The other type of memory mentioned above, Activity-based prospective memory, is most similar to delayed execute event based prospective memory. With this type of intention, it is based around the idea that when X is completed, I will do Y. As an example, when I complete washing the dishes, I will turn on television [22].

The programme will help stroke patients develop their prospective memory, by practising a number of tasks which involve prospective memory in the virtual reality environment. To make it more complicated for the patient, and to simulate real life, a number of tasks will be occurring and be interrupted and overlapped. This complexity and requiring more delayed executions of tasks, will make the system much more challenging for the patient's prospective memory [21].

2.2.2 Constructing a World

Rehabilitation is most effective in a familiar environment [23]. This is because in a familiar environment, the patient can practise activities which they will perform on a regular basis in the real world, and receive the same activation cues, when they return home. For this reason the stroke patient environment will be modelled on each patient's house, and in extreme precision. To further augment the rehabilitation process, tasks will be tailored to each patient's individual needs and tastes. Similar tasks can be employed, but with different specifics. For example, for someone who does not drink coffee, 'drink hot chocolate' may appear instead.

By the nature of the system, there will be a practical upper limit on the complexity of objects. Because of this, the tasks must be broad enough that they do not generally require a huge complexity of objects. For example, it is impossible for the system to simulate a computer that is more powerful than the computer it is running on, and impractical for it to simulate most complex computing tasks.

2.3 Problem Templates

Problem templates are a generalised solution to solving a variety of problems, and are assembled mentally by experts. They are defined in [1], as *"chunks of domain-specific knowledge, compiled mentally by experts, and used to solve commonly occurring problems in a particular domain."* This is based on the earlier concept of memory templates, such as those seen in [24].

In [24], templates were used to show how a chess expert would remember more of setup on a chess board than could be explained by the traditional model of short term memory. They discovered that these chess masters were, in fact, using long term memory retrieval and templates to recall other instances of this situation, instead of storing the location of each piece. Because this way they were storing less information in short term memory, thus, they had a much better memory of the locations. These templates apply to experts in any field, it was found.

Because experts have these problem templates in their mind, they can react to novel situations quickly, applying a template that worked in another situation, and using what they already know about similar domains to reach a solution. There are caveats to this situation, and Ohlsson [25] provides an examination of the effects of applying experience to new situations.

One of the aspects which make problem templates so flexible is their 'stackable' nature. Problem templates can consist solely of other problem templates. This allows for the easy creation of many problem templates, reusing components ranging in scope from highly granular, to big picture.

An example of a problem template would be turning a left around a corner in a car. This will in turn be made up of a number of actions, and is applicable in a variety of situations, ranging from turning left onto Bannister Place from Geneagles Terrace in a Subaru legacy, to turning left from University Drive onto Ilam Road in a Hyundai Getz.

2.4 Constraints

In the past, a number of methods have been proposed to model and support learning. These range from 'bug libraries,' to model tracing, and now to Constraint Based Models. Constraint Based Modelling was proposed by Ohlsson in [9]. Constraints are atomic individual facts, which knowledge of a domain is comprised of. They are often represented in the form "If <relevance condition> is true, then <satisfaction condition> must also be true, otherwise something has gone wrong" [26].

None the less, the extremely complex and dynamic nature of the greater stroke system, necessitates that Constraint Based Modelling be employed for describing the tasks required. Were Constraint Based Modelling not to be used, actions would have to be mapped out for every possible solution a patient could come up with for each situation. This would allow for a higher level of feedback, but take an extremely long amount of time. By contrast, a Constraints Based System only requires the definition of the relevance and satisfaction conditions. This is loosely similar to describing the states in a system, which is simple, but still facilitates quite a high level of feedback.

2.5 Dependency Graphs

The final output for the research will be represented in the form of dependency graphs. A Dependency Graph is special variety of graph. It is a directed graph which is labelled. The nodes of this graph relate to a sentence of tokens. In this grammar, each node must either be a root node, or each of its proceeding nodes must be true [27].

To render Dependency Graphs, the directed graph language *dot* was used, which is part of the Graphviz System [28]. The dot language is a description of all the sub-graphs, nodes, and edges that are in the graph. From this, the programme can output a diagraph in a picture format, as illustrated in Figures 1 and 2.

```
digraph g{
  a -> b
  a [label="hello"];
  b [label="graphviz"];
}
```

Figure 1: Snippet of dot code

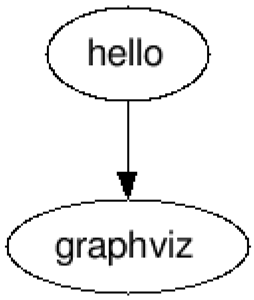

Figure 2: resultant Dependency graph

3 Design and Implementation

3.1 Goals

The overall purpose of this project is to create a system of problem templates to generate problem scenarios for the Stroke Rehabilitation System.

The main goals of this research are:

1. To evaluate the effectiveness of using problem templates;
2. To generate a number of templates which could be used in the final system;
3. To streamline the process of generating templates.

3.2 Hypothesis

The hypothesis to be tested in this research is: *Everyday tasks can be broken into components and then generalised into problem templates such that the larger stroke patient system can generate large numbers of problems.*

3.3 Deciding the Domain

To give the project a context, it was imperative that the domain be described. This includes not only deciding on a context for the project, but also defining which problem templates should be created for this research. The core reason for the research being embarked upon was to create a system for the generation of problem scenarios for the Stroke Rehabilitation System. Therefore, the domain became tasks that stroke patients could perform, which would be helpful to their rehabilitation, and which they might reasonably perform in their day-to-day life. In keeping with the goals of the larger Stroke Rehabilitation Project, it would be best for these tasks to be complex, or time related enough, so as to exercise a patients prospective memory. This could be either by the interruption of the process, or by elements within the template itself.

Selecting which tasks are to be made into problem templates is a complicated process. Much research has been done into which tasks can be performed by recovering stroke patients, and therefore, a number of scales have been formulated, such as the Instrumental Activites of Daily Living Scale (IDAL) [29] or the Barthel index [30]After conducting research into what tasks were suitable, the final six tasks were selected: *doing the dishes* (in a house with a dishwasher); *doing the ironing* (using an electrical iron); *doing the laundry* (in a house with a washing machine, and using a dryer); *making a frozen meal* (in a microwave); *making a hot drink* (in a house with a dishwasher, and using an electrical jug); and finally, *making a sandwich* (from pre-prepared ingredients).

The main reasons for choosing these six tasks is that they met the following criteria: they were well defined; they were simple enough for a stroke patient to perform, yet not arbitrary; and they could be easily simulated by the Stroke Rehabilitation System, without the need for an outside space. It is important for the tasks to be well defined. Were the domains around the task not well defined, it would be impossible to assess the validity of the problem templates. In a very poorly defined problem task, the states which the problem templates are made up of would have to be very broad, and only minimal feedback could be provided to the patient. Stroke patients find a number of even basic tasks extremely strenuous [31]. Often they suffer from a lack of motor control, cognitive aphasia, or other mental disability which would inhibit normal task performance. If patients find all the tasks extremely exhausting, or difficult to use, then the rehabilitation is of little use. Because the context of this project is for the larger Stroke Rehabilitation System, it is imperative that the templates are compatible with the system. For this reason, it is necessary that the tasks work within the capabilities of the system, such as only featuring the interior of houses, and focusing more on the interaction between objects and people or other objects, rather than internal interactions.

3.4 Preparing the Problem Templates

By definition, a problem template is compiled by an expert. Being an expert is sometimes defined as: "having, involving, or displaying special skill or knowledge derived from training or experience" [32]. While it would indeed be preferable to have professionals with many years of experience develop these problem templates, practical consideration for this research deemed this superfluous. Instead, because by nature these tasks are performed on a day-to-day basis, almost anyone might have a large pool of experience and could be considered an expert in some of these fields.

3.4.1 Composition of Problem Templates

The Stroke Rehabilitation System uses constraint-based modelling, so it was critical that the problem templates be composed in such a way that they could be converted to constraints with minimal effort. To this end, when a template is in its final form (a dependency graph, see below) it has been formulated such that, if a node's condition is true, then it should be that all the nodes 'pointing' to the node should also have their condition as true, or something has gone wrong. This is based around the constraint-based modelling system, with the node's condition being the relevance condition, and the pointing node's condition being the satisfaction condition. With this in mind, the problem templates consist of: imports for other files (*include*), the condition for the node (*situation*), links defining the edges (*goto* and *anchor*) and optional descriptors (*description* and *title*). The node's condition is always written as a state, as opposed to an action required to go between states. This is in keeping with constraint-based modelling system, as opposed to a model tracing system, which was originally employed in the Research.

3.4.2 Storage of Problem Templates

With a view towards the long term, it is imperative that these problem templates be stored in a manner which is not only easy for the Stroke Rehabilitation System to interpret, but is also easy for someone new to rapidly learn and understand. To this end, XML was selected for the process.

Originally, a proprietary system was adopted for this. This was based around plain English instructions, with command characters and instructions appended at the end. It started off very simple, but rapidly grew far too complex for it to be easily understood or interpreted when it was required to reference multiple files with parameters and special edges. This was further hampered by the large amount of 'hard coding' and reserved phrases. Figure 2 illustrates an example in an early phase:

[in any order]:

(2),(3),(4),(5)

2. open instant coffee

3. pick up spoon

4. use spoon to put one spoonful of instant coffee into cup.

5. put down spoon

Figure 3: source from the original system

Figure 4 shows the same example in the final form.

INCLUDES:

clean.txt:l

ingredients out.txt:i

INCLUDED

open chocholate[chc:1][jg:2]

pick up spoon [spn:1]

use spoon to put one spoonful of instant coffee into cup

put down spoon

Figure 4: source from late in the development of the proprietary system

XML was selected as a replacement. The factor which makes XML most useful for this situation is its tags which provide a context for the data. For example, having <goto>4</goto> is much easier to understand for a human than #4#. Additionally, by using a tag system, it reduces the number of reserved characters. As a well-established standard, there is a number of parsers and ancillary systems already available for XML, which makes it easier to set up in other systems. Also due to its well established nature, it is likely that anyone newly introduced to the system will have a good understanding of how to use XML. Because it is extensible, yet it can be defined in a schema or Document Type Definition, it is easy for novices to create new documents, as the domain of possible commands is well defined. Snippets of how the code could look in XML are shown in Figure 5.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Statelist SYSTEM "statelist.dtd">
<Statelist>
	<title>The Sandwich is prepared, and the area clean</title>

	…

<include>
```

```
        <file>newGetOut.xml</file>
        <friendlyName>getOut</friendlyName>
        <inst>%location%</inst>
</include>

        …

        <State>
                <goto system="breadOut" priority="do second">
                        1
                </goto>
                <goto system="getOut" inst="%location%">
                        1
                        <item>%item%</item>
                        <storage>%location%</storage>
                        <workspace>bench</workspace>
                </goto>
                <anchor>1</anchor>
                <description>Items are out</description>
        </State>
…
</Statelist>
```

Here's a snippet showing a 'situation':

```
                <situation>
                        <object>
                                <holdable>knife</holdable>
                        </object>
                        <prop>
                                <heldby>actor[1]</heldby>
                        </prop>
                </situation>
```

Figure 5: The raw XML for a problem template

3.4.3 Presentation of Problem Templates

In order for the expert to be able to assess the problem templates, it was required that they be outputted in the format of a dependency graph, as seen in Appendix 1.

By using dependency graphs, it is easy to see the relation between nodes. This makes it easy for a human to read and interpret which conditions need to be met, to meet other conditions. For example, for the jug to be on, it should be that there is water in the jug, otherwise something has gone wrong. This is clear in the graph, as a user can just look backwards along the arrows. Graphs are only displayed for problem templates which have been instantiated with a parameter. For example: hot drink (coffee), instead of just hot drink.

3.5 Systems to Streamline the Process

From the very beginning, it has been very important that the system be constructed in such a way that it will be very easy and efficient for future users to add problem templates to the system. One way in which this was done, was by writing a programme to allow for the easy conversion of XML files into dependency graphs. This programme, called $\frac{cx}{dt}$ (convert XML to dependency tree), allows a user to easily produce problem trees through a command line interface, with the programme accepting a file name for the problem template, and variables to fill in the parameters. The programme itself outputs the data in a format known as dot, which can be interpreted by Graphviz [28] to produce the output graphs in an SVG format. With just $\frac{cx}{dt}$ and graphviz, a simple batch file can process hundreds of problem templates and parameters into picture files, ready for assessment or demonstration.

3.6 Tools used

For the generation of the XML files: Eclipse, Notepad++, and Oxygen XML editor were used.

For rendering the problem scenarios, Graphviz Workspace, ZGRViewer, and the online tool Graphviz [28] were used.

For writing the programme, Java and Eclipse were used.

4 Evaluation

4.1 Description

To assess the usability of the Problem Templates, a domain expert inspected a number of individual problem scenarios, generated from the templates and assessed their correctness. To best simulate a real-world usage scenario, data was sourced from a diverse range of real people, and then compiled with the Problem Templates, into a number of problem scenarios which were assessed. This assessment of whether the problem scenarios generated from the Problem Template is the determinant of if the hypothesis holds.

Over several months, the problem templates were compiled. This was an iterative process, some graphs having dozens of iterations in the prototyping stage. Basic research was done to ascertain standard ways of conducting these processes. Six different problem templates were made, covering a range of tasks that can be performed by problem templates.

Around 40 individuals, of diverse background and taste, were asked up to 7 questions: their favourite powder based instant hot drink, which ingredients do they put in the drink, What would be their ideal ingredients of a sandwich, what is the process they use to wash clothes, and what is added to the washer during the process, How do they get water into their iron (if applicable), What is their favourite frozen meal, and finally, Other than dishes, what do they put in a dishwasher. This information would provide usage scenarios for each of the problem templates. After duplicate and invalid answers were removed from the pool of answers, the data was parsed into a batch file, and the answers converted into arguments for the programme. This process, as in 3.6, produced a number of output graphs. In all, 76 output graphs were produced and submitted for assessment. These 76 graphs represented individual problem scenarios for each responder to the survey, and simulated how the program would work as part of the bigger system, which each responder playing the part of the patient.

The Domain Expert inspected the validity of each graph. After he had examined each and every one, he arrived at a set of conclusions around the results. Because the expert is also involved in the stroke project, he was also able to provide insight into the applicability of these files towards the stroke rehabilitation effort.

4.2 Results

The main finding was that on the whole, there were no fundamental changes required for the problem scenarios to be valid. That being said, not all problem scenarios were in fact valid. This was largely due to typographical or minor errors, which were not significant enough to affect the overall validity of the concept.

The expert did however, recommend that to improve the modelling, and to make more compatible with systems, changes should be made to the templates. The number and variety of specialised edges should be reduced. Instead of including an ordering on some edges, (for example: ‘do first’), a re-arrangement of nodes, and an additional edge would in most cases suffice, and require less interpretation from the system. Furthermore, to improve readability, and to make the system more ‘solid’, disjunctions should be used. This would entail an extra node, and make it easier to process cases involving XORs options located amongst OR.

All in all, these results imply that the Hypothesis holds.

4.3 Discussion

Throughout this evaluation, we were able to create a number of problem templates to simulate a large variety of activities that could be performed by stroke patients. After research, as well as advice from medical professionals, and stroke patient carers, a shortlist was compiled outlining the large portion of the tasks which are possible to simulate, be performed by stroke patients, and would provide a useful activity for them. Because this list is quite a small domain, it was actually possible to cover a large portion of this domain in the project. As a result of this we can say with some degree of confidence that these findings would apply to most if not all of the relevant tasks- that is, tasks which can be performed by a stroke patient and simulated in the Stroke Rehabilitation Project.

More than seventy groups of specific variables were entered into the system, to create problem scenarios. This is a reasonably large trial of its type. To provide a point of contrast, [8] only used 49 bugs, to test its dynamically generated, template based testing suite. Ideally an even larger follow-up trial could be embarked upon, although the benefits of doing so could be limited. One interesting trend in the dataset was the number of duplicates and invalid responses. 229 responses were captured, but more than 66% of these responses were either substantially duplicated elsewhere (e.g. 'sugar and milk' and 'milk and sugar') or invalid for the question (e.g. 'Tea' as an instant hot drink, whereas tea normally takes time to stew in the teapot).

As mentioned elsewhere in this report; one benefit of the tasks being of an everyday nature was that most people are of a level of experience akin to that of an expert in other fields. As a result of this, the expert was able to ascertain not only the validity of the templates, but also of their usefulness within the context of stroke rehabilitation system. If the study was to be performed again on a larger scale, it would be beneficial to have a larger review panel, to ensure higher accuracy. However, the marginal benefits of these extra judges would likely low, and their roles largely redundant.

The expert drew a number of conclusions about the system. Of the minor errors, there were a few types. A few were situations where the node had been labelled with actions (for example: 'clean up items') instead of a similar state they were intended to be labelled as ('the items have been cleaned up'). Other minor errors included one node that was labelled 'washing.heldby =washing.heldby'. These types of errors are attributable to human error, and not as a result of the system.

Another area of improvement the expert pointed out was the use of additional edges in place of many of the specialized edges. Specialised edges (such as 'do first') were implemented to make the systems easier to understand and reduce cyclomatic complexity. This change is a change that will be implemented with relative ease. The Stroke Rehabilitation System could still be set up in such a way to interpret the graphs with either system, so this is not a prohibiting factor in validation of the hypothesis.

Overall, according to this assessment, the system requires no fundamental changes to be able to complete its goals of problem generation using Problem Templates for the Stroke System. This is a strong endorsement of the hypothesis, and validates the arguments of this research.

5 Conclusion and Future Direction

5.1 Conclusion

A system such as the envisioned Stroke Rehabilitation System (outlined in section 2.2) would require a strong engine behind it, to generate problem scenarios for the patients. Were this not the case, the systems usefulness would be impaired, either by the difficulty in generating problem scenarios, or by the patient being forced to work in a more alien environment.

The use of Problem Templates (as explained in section 2.3), allows for the rapid creation of flexible problem scenarios, tailored to the patient's needs and preferences. Because Problem Templates are discrete and reusable, as more are created, it will become easier to generate additional high-level templates, as the low-level templates may already be available for use.

Section 3 outlined the hypothesis for the project, as well as the three key goals for the process. The first goal was to evaluate the effectiveness of Problem Templates in this context. This was completed to a degree. The effectiveness was assessed, even if in a slightly limited capacity. It was found that they were valid, and would function in their intended role. The second was to complete a number of Problem Templates which could be used in the final system. Some of the Problem Templates are suitable in the current state, although for many of the top-level Problem Templates, refactoring will be required to ensure suitability. The last of the goals was to streamline the creation of Problem Templates. The process is now at the point where Problem Templates can be generated in a number of hours, as opposed to the weeks it took to craft the first one to a satisfactory level. This constitutes a major optimisation of the creation process.

The Hypothesis appears to be supported by the evidence supplied in the assessment. This is encouraging news, and hopefully will encourage more research into, and production of, Problem Templates in the context of stroke rehabilitation.

5.2 Future Direction

If this research is to be continued, the first step would be to implement the changes recommended in the assessment conducted by the Expert (sections 4.3 and 4.4). To begin this, a system of disjunction nodes would be added to the Problem Templates, reinforcing the division between XOR and OR. Node ordering edges would be replaced with normal edges, and additional edges added, to enforce an order. Additionally, changes would be done to fix the minor errors, throughout the Problem Templates.

The next phase would be to extend the number of Problem Templates. The next goal would be around fifteen templates to provide a broader view. This would need to be done with the assistance of medical professionals, to ensure that the added tasks add value to a training regimen. Following that, a larger study could be conducted, with a formal and in-depth survey to ensure that the system, works on a larger scale.

Assuming later studies confirm the results identified here, eventually the system would be integrated with the stroke rehabilitation process to allow procedural generation of problem scenarios.

6 Bibliography

[1] M Mathews and A Mitrović, "Investigating the Effectiveness of Problem Templates on Learning in Intelligent Tutoring Systems," in *Proc, 13th Int. Conf. Artificial Intelligence in Education AIED 2007*, Los Angeles, 2007, pp. 611-613.

[2] P Kader et al., "Stroke: Awareness of the Signs, Symptoms and Risk Factors - A Population-Based Survey," *Cerebrovascular Diseases*, vol. 16, no. 2, p. 134, 2003.

[3] C Anderson, J Linto, and E Stweart-Wynne, "A Population-Based Assessment of the Impact and Burden of Caregiving for Long-term Stroke Survivors," *Stroke*, vol. 26, pp. 843-849, 1995.

[4] B Johansson, "Brain Plasticity and Stroke Rehabilitation," *Stroke*, vol. 31, pp. 223-230, 2000.

[5] G Kwakkel, R Wagenaar, T Kielman, G Lankhorst, and J Koetsier, "Effects of Intensity of Rehabilitation After Stroke," *Stroke*, vol. 28, pp. 1550-1556, 1997.

[6] A Maujean, D Shum, and R McQueen, "Effect of Cognitive Demand on Prospective Memory in Individuals with Traumatic Brain Injury," *Brain Impairment*, vol. 4, no. 2, pp. 135-145, 2003.

[7] F Staub and J Bogousslavsky, "Fatigue after Stroke: A Major but Neglected Issue," *Cerebrovascular Diseases*, vol. 12, no. 2, pp. 75-81, 2001.

[8] W Tsai, Y Tu, W Shao, and E Ebner, "Testing extensible design patterns in object-oriented frameworks through scenario templates," in *Computer Software and Applications Conference, 1999. COMPSAC '99. Proceedings. The Twenty-Third Annual International* , Phoenix, AZ, USA, 1999, pp. 166-171.

[9] S Ohlsson, "Constraint-Based Student Modelling," in *Student Modelling: The Key to Indicidualized Knowledge-Based Instruction.*: Springer-Verlag, 1994.

[10] M Brown, "Brain attack: a new approach to stroke," *Clinical Medicine, Journal of the Royal College of Physicians*, vol. 2, no. 1, pp. 60-65, 2002.

[11] NR Sims and H Muyderman, "Mitochondria, oxidative metabolism and cell death in stroke.," *Biochim Biophys Acta*, vol. 1802, no. 1, pp. 80-91, Jan 2010.

[12] P Pedersen, H Jørgensen , H Nakayama, H Raaschou, and T Olsen, "Aphasia in acute stroke: Incidence, determinants, and recovery," *Annals of Neurology*, vol. 38, no. 4, pp. 659-666, 1995.

[13] R Bonita and R Beaglehole, "Recovery of motor function after stroke," *Stroke*, vol. 19, pp. 1497-1500, 1988.

[14] V Mok, "Cognitive impairment and functional outcome after stroke associated with small vessel disease," *J Neurol Neurosurg Psychiatry*, vol. 75, pp. 560-566, 2004.

[15] G Kinsella, B Ong, and J Tucker, "Traumatic Brain Injury and Prospective Memory in a Virtual

Shopping Trip Task: Does It Matter Who Generates the Prospective Memory Target?," *Brain Impairment*, vol. 10, no. 1, pp. 45-51, 2009.

[16] H Roediger, "Prospective Memory and Episodic Memory," in *Prospective Memory Theory and Applicaitions*. Mahwah, New Jersey: Lawrence Erlbaum Associates, Publishers, 1996.

[17] A.J. Sellen, G Louie, J.E. Harris, and A.J. Wilkins, "What Brings Intentions to Mind? An In Situ Study of Prospective Memory," vol. 4, no. 5, pp. 483-507, 1997.

[18] D Shum, M Valentine, and T Cutmore, "Performance of Individuals with Severe Long-Term Traumatic Brain Injury on Time-, Event-, and Activity-Based Prospective Memory Tasks," *Journal of clinical and experimental Neuropsychology*, vol. 21, no. 1, pp. 49-58, 1999.

[19] J.L. Mathias and K.M. Mansfield, "Prospective and declarative memory problems following moderate and severe traumatic brain injury," *Brain Injury*, vol. 19, no. 4, pp. 271-282, 2005.

[20] Yvonne Groot, B Wilson, J Evans, and P Watson, "rospective memory functioning in people with and without brain injury," *Journal of the International Neuropsychological Society*, vol. 8, pp. 645-654, 2002.

[21] M McDaniel, G Einstein, T Graham, and E Rall, "Delaying execution of intentions: Overcoming the costs of interruptions," *Applied Cognitive Psychology*, vol. 18, pp. 533-547, 2004.

[22] G Brewer et al., "A comparison of activity-based to event-based prospective memory," *Applied Cognitive Psychology*, vol. 25, no. 4, pp. 632-640, 2011.

[23] R McNeny, "Therapy for Activities of Daily Living: Theoretical and Practical Perspectivies," in *Brain injury medicine: principles and practise*. New York: Demos Medical Publishing, 2007.

[24] F Gobet and H A Simon, "Templates in chess memory: A mechanism for recalling several boards," *Cognitive Psychology*, vol. 31, no. 1, pp. 1-40, 1996.

[25] S Ohlsson, "Learning from performance errors," *Psychological Review*, vol. 103, no. 2, pp. 241-262, 1996.

[26] A Mitrovic, K Koedinger, and B Martin, "A Comparative Analysis of Cognitive Tutoring and Constraint-Based Modeling," in *Lecture Notes in Computer Science*. Berlin: Springer Berlin / Heidelberg, 2003, vol. 2702, pp. 313-322.

[27] J Nivre, *Inductive Dependencey Parsing*. Dordrecht, The Netherlands: Springer, 2006.

[28] E Gansner, K Rleftherios, and S North. (2009, December) Documentation | Graphviz - Graph Visualization Software. [Online]. http://www.graphviz.org/Documentation/dotguide.pdf

[29] M.P. Lawton and E.M. Brody, "Assessment of Older People: Self-maintaining and instrumental activities of daily living," *Gerontologist*, vol. 9, pp. 179-186, 1969.

[30] FI Mahoney and DW Barthel, "Functional evaluation: the Barthel index," *Maryland state medical journal*, vol. 14, pp. 61-65, 1965.

[31] JC Swieten, PJ Koudstaal, MC Visser, HJ Schouten, and J van Gijn, "Interobserver agreement for the assessment of handicap in stroke patients," *Stroke*, vol. 19, pp. 604-607, 1988.

[32] Merriam Webster. Merriam-Webster online. [Online]. www.merriam-webster.com/dictionary/expert

7. Appendix 1:

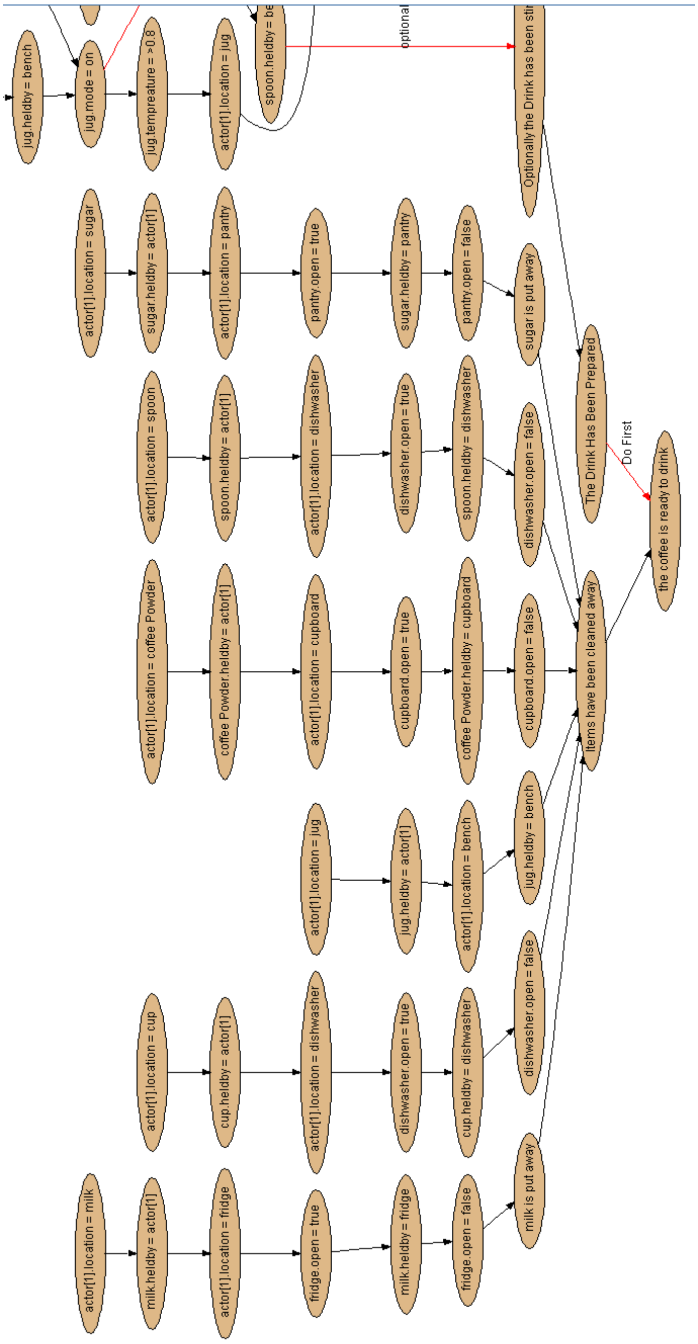

Figure 6: the 'clean-up' portion of the Coffee-make instant hot drink- dependancy graph

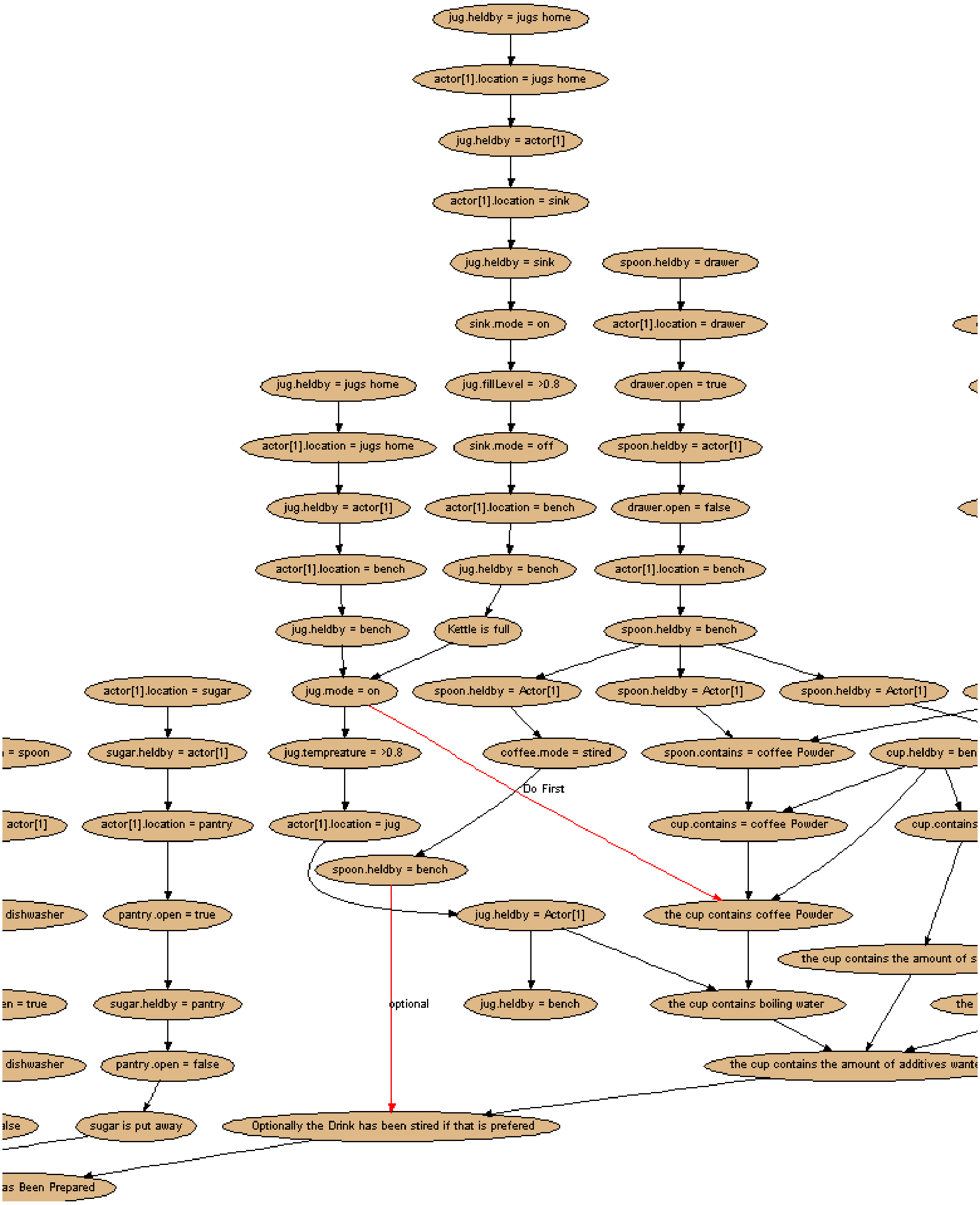

Figure 7: the boiling water and coffee powder section of the graph, notice the overlap to the left with figure6, and to the right with figure 8

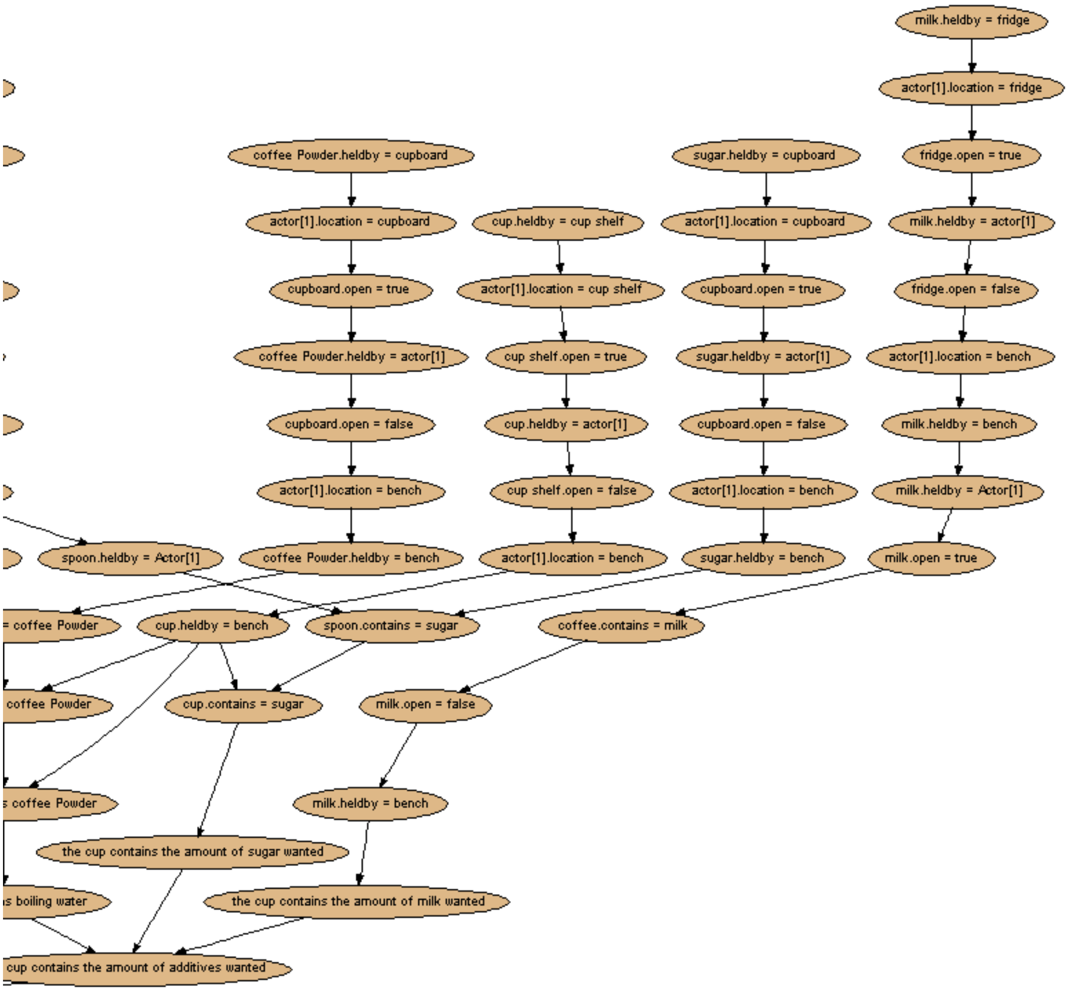

Figure 8 the 'ingredients out' portion of the Coffee-make instant hot drink- dependency graph