Model-driven Development of Mobile Personal Health Care Applications

Abizer Khambati, John Grundy, James Warren and John Hosking

Department of Computer Science and Department of Electrical and Computer Engineering
University of Auckland, Private Bag 92019, Auckland, New Zealand
akmail049@gmail.com, {john-g, jim, john}@ cs.auckland.ac.nz

Abstract

Personal health care applications on mobile devices allow patients to enhance their health via reminders, monitoring and feedback to health care providers. Engineering such applications is challenging with a need for health care plan meta-models, per-patient instantiation of care plans, and development and deployment of supporting web and mobile device applications. We describe a novel prototype environment for visually modelling health care plans and automated plan and mobile device application code generation.

1. Introduction

Research shows that current health care systems cater well for patients suffering from acute illnesses but are much less suited to patients suffering from long term chronic illnesses [1]. Technological solutions for providing support to chronically ill patients and their health care providers are numerous. Visual modelling suites such as Protégé and Asbru [5][9] give health care providers the ability to digitally represent clinical guidelines. Others, such as Logician, allow health care providers to model and schedule patient's tasks and activities [9]. These systems aid health care providers by providing them ways to reuse health information without concern for how this supports the patient in proactive disease management.

Several approaches support the patient in managing their health. The weight wellness monitor (WWWM) [10] provides a wireless architecture to monitor and manage weight loss in overweight individuals. It focuses on capturing weight data from bluetooth devices to a "home health server". The Patient centred application mobile energy balance (PmEB) [11] is aimed at people with eating disorders which lead to them being overweight or obese. The web-based My Health, My Life (MHML) [7] wellness application is aimed at patients suffering from various long term illnesses such as diabetes, high blood cholesterol, or obesity. These solutions focus on how health care information is represented to patients for support but lack concern for how this information is captured or structured.

We have developed a new approach to modelling care plans for chronic disease using two domain-specific visual languages (DSVLs). The first allows health care providers to model complex care plans, health activities, performance measurements and sub-care plans. The second DSVL describes a mobile device interface for the care plan. A code generator synthesizes a mobile device implementation of this care plan application.

2. Motivation

Consider the case of John Doe, a 35 year old engineer who suffers from class 1 obesity. Because of his condition he faces many risks: e.g. diabetes, high cholesterol problems, or potential heart problems. John decides to see his health care provider to take control of his condition. After John's health care provider analyses his condition he decides to design a care plan for John containing: general health advice; performance goals e.g. reduce weight and improve eating habits; health care activities e.g. exercise, monitor blood pressure and weight; and assessment tips e.g. amount of donuts eaten today less than target. However it is hard for John to adhere to such a complex care plan. He has trouble gauging his progress and easily becomes disheartened. His health care provider can not easily monitor how well John is doing against his plan.

In an earlier project we developed a prototype mobile health care application for chronic disease management (e.g. diabetes, asthma, obesity) [4]. As part of this work we developed a preliminary conceptual model for representing such health care plans and storing them in an XML format for reuse. Developing these care plan models and the mobile-hosted care plan applications, however, proved hard. Health Care providers could not directly author the UML-based care plan models due to the low *closeness of mapping* [2] of the modelling language used. Instantiating the care plan required complex editing of XML-based models. Developing a care plan application with .NET Compact produced a "nice" interface but customisation for other care plans and patients requiring much coding.

We wanted to provide health care providers with a more accessible modelling language to develop their care plans and to fully generate code for a mobile care plan application from the models. In our approach a health care provider (doctor, nurse, specialist) develops one or more care plans using a domain-specific visual language (DSVL) tool based on an extension to our earlier generic health care plan model. Our health care plans comprise structured information about health advice, performance targets, care activities, assessment/evaluation approaches and possibly "sub-care plans" to enable structuring of complex care plans. For example the Obesity Care plan shown in Figure 1 has a number of sub-care plans e.g.

978-1-4244-2188-6/08/$25.00 ©2008 IEEE

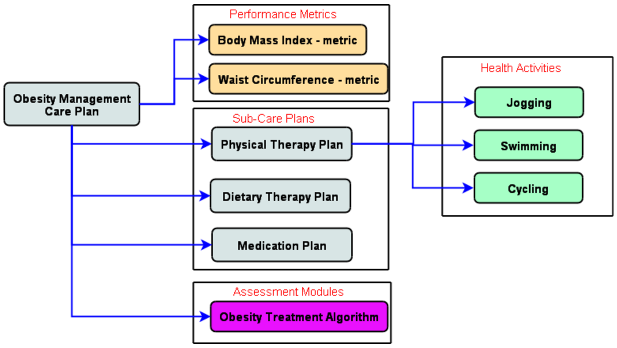


Figure 1. High-level health care example care plan.

physical, dietary and medication, some of which may be shared with other plans.

These care plans are stored and a reused by "instantiating" (customizing) them for a particular patient. Typically this means setting appropriate dates, times, medication amounts, exercise durations and timings, performance measurement targets, and deleting inappropriate sub-care plans. A trained visualisation modeller then develops one or more visualisation specifications using a second DSVL. This describes how sets of items in a care plan model are related to visual elements – screens, text boxes, labels, dates, edit boxes, radio buttons etc. This DSVL is "platform neutral" i.e. does not depend on a particular implementation technology to realise the care plan interfaces. Multiple visualisation specifications can be defined for the same health care plan as well as patient specific visualisations. The instantiated patient care plan and chosen visualisation are used to synthesize a care plan application. Currently we generate an OpenLaszlo implementation [8] which is compiled dynamically to Flash Shockwave or DHTML/ JavaScript for rendering in a web browser. Web services read/write patient information for viewing by a health care provider.

3. Example Usage

Our VCPML (Visual Care Plan Modelling Language) was designed for health care providers to capture health treatment and management information commonly contained in guidelines for chronic illness treatment into a more formal, structured and digital manner. Figure 2 (left) shows a generic obesity management plan using VCPML. This care plan (centre Care Plan icon) is composed of five sub care plans (right hand side), an obesity treatment algorithm (an assessment module, top) two performance metrics (targets for BMI and waist measurements, left hand top) and an obesity review activity (3 monthly requiring a nurse and consulting room to perform, bottom).

This obesity care plan was constructed by referring to the common treatment procedures outlined in an obesity management treatment guideline [6]. Figure 2 (right) elaborates the Physical Therapy sub care plan. into a more detailed view. Two extra activities are defined, Jogging and Swimming (the conceptual schema is recursive; sub care plans can have the same components as care plans). Each has a set of instructions specifying warm up actions and a routine schedule. The Obesity Treatment assessment module also encapsulates a visually specified algorithm for deciding on treatment options. This consists of various assessment activities, choice points and treatment recommendations in a flow chart formalism (not shown) that mirrors the obesity guideline's notation [6].

Using VCPML the health care provider can design care plan templates. However the care plan model at this stage is a generic one which can be applied to any patient. Instantiation personalises for a particular patient. This includes discussing with the patient their lifestyle and their daily routines. Then, using our "care plan instantiation tool", the health care provider can load a VCPML care plan template and: add care plan components to match the patient's needs, such as extra health care activities or care plans, through an inbuilt wizard; de-select components of the plan; and/or modify the activity schedules specified.

An instantiated health care plan contains the information needed for a patient to self manage their chronic ailment. However this information needs to be provided to the patient in an accessible way. The next step therefore is to design a mobile device application for the patient to graphically present the care plan instance to the patient, and allow the patient to interact with it. This uses a second DSVL, VPAM (Visual Patient Application Modeller) to map the data structures implied by a VCPML care plan to high level GUI interaction models. VPAM allows the user to define the look and feel of a patient application, by describing how each part of the care plan instance will be graphically represented by GUI elements. An application generator generates the patient application in OpenLaszlo [8]. The application can then be compiled to FLASH or DHTML. These formats were chosen as they run on most mobile devices (particularly FLASH), so are largely device and platform independent.

Figure 3 (left) is a VPAM model for the main screen of John's care plan application. This specification maps elements from a care plan repository (specifically names of available care plans) to a navigation list on the main screen. The generated Flash screen (with just an obesity plan loaded) overlays the figure. Figure 3 (right) shows a more complex screen specification and generated form to capture a Blood Glucose measurement. This extracts details from an Appointment data element, arranges the fixed values as labels, the data value to capture as a textbox and a button that causes the response to be saved.

4. Design and Implementation

We designed and developed our two domain-specific visual languages using the Marama meta-tools [3]. We translated our generic care plan meta-model into a more detailed Marama meta-model describing care plan components and their properties. Visual shapes to

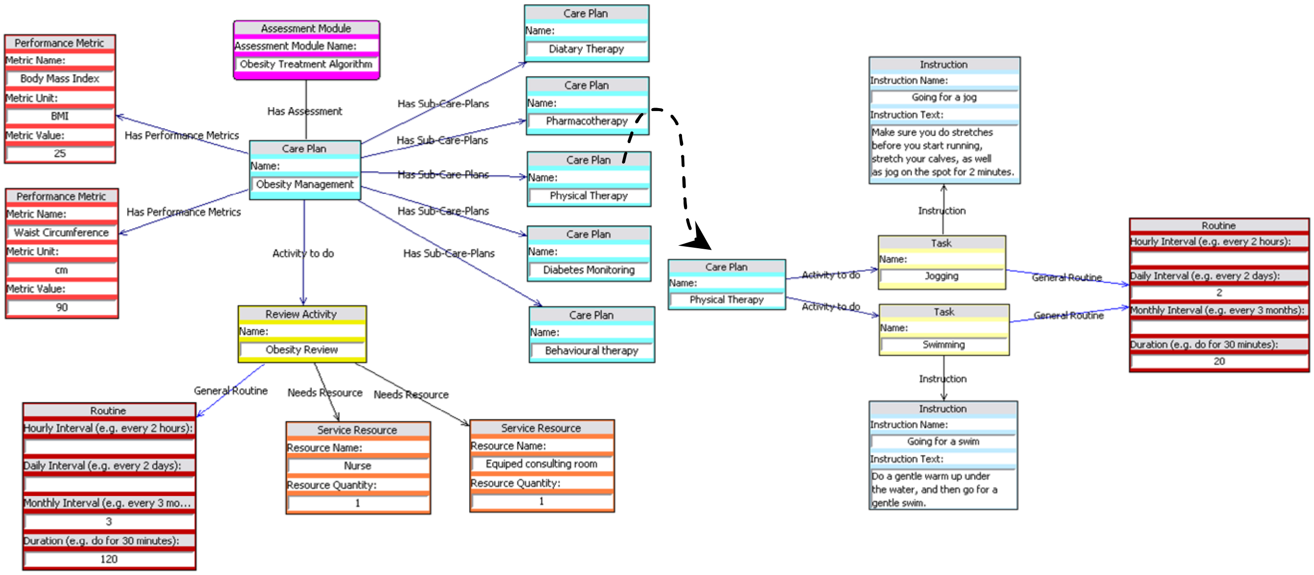
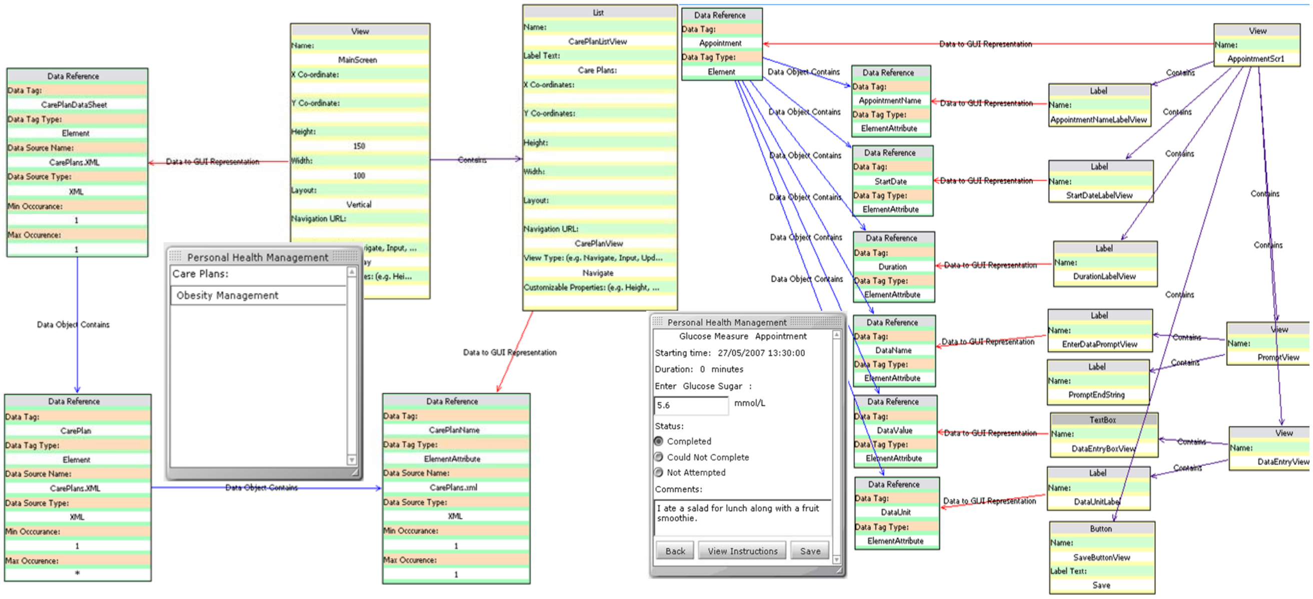


Figure 2. VCPML models of an obesity treatment plan.



Personal Health Management

Care Plans:

Obesity Management

Personal Health Management

Glucose Measure Appointment

Starting time: 27/05/2007 13:30:00

Duration: 0 minutes

Enter Glucose Sugar :

5.6 mmol/L

Status:

● Completed

○ Could Not Complete

○ Not Attempted

Comments:

I ate a salad for lunch along with a fruit smoothie.

Figure 3. VPAM describing parts of mobile patient application user interface and two example generated interfaces.

represent care plan components in the meta-model were then designed and mappings between the components in the care plan meta-model and their respective shapes specified. The Marama meta tool then generated the care plan DSVL as an Eclipse plugin. Health care plans modelled in the DSVL are saved to XMI and then transformed via XSLT scripts into a custom health care plan XML format we have developed. The care plan instantiation tool is a .NET application that loads a selected care plan XML model and allows the health care provider to tailor it for specific patient. The VPAM tool is used to design a mobile application interface for a care plan model. Our application generator takes the instantiated patient care plan, VPAM interface specification, and generates an OpenLaszlo implementation using XSLT.

5. Evaluation

We have carried out two evaluations of our approach. A Cognitive Dimensions [2] (CD) analysis assessed the visual languages and support environment. An informal user evaluation assessed the care plan visual modelling language, application interface modelling language and overall approach to modelling and generating health care applications. VCPML provides good *closeness of mapping* for the target end users. Despite dealing with fairly abstract concepts the abstractions present (both for the care plans and the algorithm specifications) are relatively familiar to the target end users so VPCML has a low *abstraction gradient*. The language is *terse* (only 12 different symbols) so there are not a large number of concepts to learn. As with most box and connector diagrams, VCPML suffers from *viscosity* problems (inability to easily modify things) as layouts usually need manual rearrangement to fit in additional components. The use of multiple views alleviates the viscosity problem but introduces *hidden dependencies*. We feel that these are manageable as they do not involve cross cutting relationships, just elaborations with a single point of connection. A number of other minor *hidden dependencies* arise in VCPML between resources, routines and instructions, and ownership of sub-care plan elements, but these are relatively minor in our opinion.

VPAM is more of a work in progress. It has a lower *closeness of mapping*: the notation used for the interaction components is quite abstract and by no means WYSIWIG. The *abstraction gradient* for this target group is low though as the abstractions are all familiar. The language is more *verbose* than VCPML as it needs to represent a large range of widget types. The relative similarity between the iconic forms is thus problematic so VPAM is likely to be more *error prone.* VPAM has similar *viscosity* and *hidden dependency* problems to VCPML.

The informal user evaluation provided initial feedback on the usability of the care plan modelling system and the value of the general approach. The user group consisted of three experienced software developers. The first two sections of the evaluation provided users with VPCML and VPAM modelling scenarios. The users were asked questions on their understanding of the DSVL models. All participants were able to answer all the questions related to VCPML successfully, showing that the care plan and assessment module models can be easily read and understood and with ease of understanding of the VCPML models rated as 8 out of 10. The main deficiencies noted related to hidden dependencies, in line with the CD analysis. Participants could also answer all questions related to the VPAM models. However all found this graphical notation to be inadequate to the modelling task due to its abstractness; the rating for ease of understanding was 6 out of 10, also in line with the CD analysis.

The last evaluation section focussed on the overall approach. Participants rated it as being innovative; all participants agreed the underlying concept was a good one and rated the overall approach at 7 out of 10. Advantages for health care providers noted were that the approach supported a "collaborative" and "proactive" approach as patients and health care providers together design patient health care strategies. Participants suggested that the approach allows reuse of health care plans and is useful in transferring some responsibility of care to the patient. These responses were in line with our original motivations for developing the care plan modelling system.

Patient advantages were that it "passes responsibility" to the patient, by converting a static care plan into something "operational", "executable" directly related to the patient, keeping the patient connected to the care plan. Participants indicated that using mobile device deployment was a good idea as care plans are made portable. However, they noted that much care is needed to ensure that technologies used are fully tested and error free.

A more complete evaluation with the target end user groups together with the cognitive dimensions analysis provides us with sufficient indicative feedback to suggest that the approach we have taken is a sensible one with sufficient merit to develop beyond the proof of concept system reported here. It is clear that while the conceptual VPAM model is adequate enough its surface formalism requires additional development. However, prior to redeveloping this formalism our immediate focus will be on using the existing formalism on a much broader range of example care plans. Our feeling is that this will suggest a set of higher level more "business oriented" primitives for designing the GUI elements of patient applications and that these higher level primitives will form the basis for the revision of the VPAM formalism.

References

[1] Alaoui, A., et al. Diabetes Home Monitoring Project. in Medical Technology Symposium. 1998. Pacific.

[2] Green, T.R.G. & Petre, M. Usability analysis of visual programming environments: a 'cognitive dimensions' framework. JVLC 1996 (7), pp. 131-174.

[3] Grundy, J.C., Hosking, J.G., Huh, J. and Li, N., Marama: an Eclipse Meta-toolset for Generating Multi-view Environments, ICSE 2008, Leipzig, Germany, 2008, ACM.

[4] Khambati, A. and S. Kanade, Care Plan Modelling Framework, University of Auckland Part 4 Software Engineering Project Reports, 2006.

[5] Miksch, S., Y. Shahar, and P. Johnson, Asbru: A Task-Specific, Intension-based, and Time-oriented Language for Representing Skeletal Plans, in 7th Workshop on Knowledge Engineering: Methods & Languages. 1997: Milton Keynes, United Kingdom. p. 25.

[6] National Institute Of Health, N.H., Blood and Lung Institute, Clinical Guidelines On the Identification, Evaluation, And Treatment Of Overweight And Obesity In Adults. 1998. p. 228.

[7] Nuschke, P., T. Holmes, et al. (2006). My Health, My Life: A Web-Based Health Monitoring Application. Human Factors in Computer Systems. Montreal, Quebec, ACM.

[8] OpenLaszlo, 2007: http://www.openlaszlo.org/.

[9] Open Clinical: Knowledge Management for Medical care, Guideline Modelling Methods and Technologies, 2007, http://www.openclinical.org/gmm_protege.html.

[10] Pärkkä, J., et al. A Wireless Wellness Monitor for Personal Weight Management, Information Technology Applications in Biomedicine. 2000.

[11] Tsai, C. c., G. Lee, et al. Usability and Feasability of PmEB: A mobile phone applicaiton for monitoring real time caloric balance, Mobile Network Applications 12: 173-184.