UNIVERSITY OF AMSTERDAM

UvA-DARE (Digital Academic Repository)

Automated dialogue generation for behavior intervention on mobile devices

Fitrianie, S.; Griffioen-Both, F.; Spruit, S.; Lancee, J.; Beun, R.J.

DOI
10.1016/j.procs.2015.08.339

Publication date
2015

Document Version
Final published version

Published in
Procedia Computer Science

License
CC BY-NC-ND

Link to publication

Citation for published version (APA):
Fitrianie, S., Griffioen-Both, F., Spruit, S., Lancee, J., & Beun, R. J. (2015). Automated dialogue generation for behavior intervention on mobile devices. *Procedia Computer Science*, *63*, 236-243. https://doi.org/10.1016/j.procs.2015.08.339

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, P.O. Box 19185, 1000 GD Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (https://dare.uva.nl)

Download date:09 Sept 2026

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 63 (2015) 236 – 243

Procedia Computer Science

The 5th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH 2015)

Automated Dialogue Generation for Behavior Intervention on Mobile Devices

Siska Fitrianie[a]*, Fiemke Griffioen-Both[a]†, Sandor Spruit[a], Jaap Lancee[b], Robbert Jan Beun[a]

[a]*Information and Computing Sciences, Utrecht University (UU), PrincetonPlein 5, 3584CC Utrecht, the Netherlands*
[b]*Dept. of Clinical Psychology, University of Amsterdam (UvA), Weesperlein 4, 1018XA Amsterdam, the Netherlands*

Abstract

Communication in the form of dialogues between a virtual coach and a human patient (coachee) is one of the pillars in an intervention app for smartphones. The virtual coach is considered as a cooperative partner that supports the individual with various exercises for a behavior intervention therapy. To perform its supportive behavior, the coach follows a certain interaction model and its requirements, such as alignment, mutual commitment and adaptation. In this paper, we propose E-Coach MarkUp Language (ECML), a standard XML specification for scripting discourses that define how the virtual coach maintains a dialogue with a coachee following the interaction model. The format of the language allows messages to be tailored at a fine-grained level. Each sentence is synthesized based on the inferred goals of the coaching process and the current beliefs of the user, incorporating everything that has been said previously in the conversation. The design enables inexpensive implementation on mobile devices for a flexible, seamless coaching dialogue. With expert-based evaluations, we validated the language using scenarios on implemented ECML in the field of insomnia therapy.

© 2015 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
Peer-review under responsibility of the Program Chairs

Keywords: Dialogue generation, mobile app, cognitive behavior therapy

* Corresponding author. Tel.: +31-30-2537886; fax: +31-30-2532804
E-mail address: s.fitrianie@uu.nl
† *E-mail address:* f.griffioen-both@uu.nl

1877-0509 © 2015 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
Peer-review under responsibility of the Program Chairs
doi:10.1016/j.procs.2015.08.339

1. Introduction

Since Weizenbaum's Eliza in 1965, many attempts have been done to build systems that mimic a natural conversation between health providers and patients. These health dialogue systems aim at providing widely accessible and low cost healthcare in a limited treatment domain[1,2], and many of them are successfully applied in clinical trials[3,4,5]. Some of these systems have been designed for supporting people in the process of behavior change[5,6,7]. So far, the application of such systems on mobile devices seems still in its infancy.

In this work, we propose a method for automated dialogue generation in a coaching system that offers a Cognitive Behavior Therapy (CBT) program on smartphones. CBT is a short-term, focused approach to the treatment of many types of emotional, behavioral and psychiatric problems. We refer to this mobile system as the e-coach and to a CBT patient as the coachee. The e-coach is generic in nature and aims at reusability in various CBT domains. However, in this paper, we take concrete examples from the domain of insomnia therapy (CBT-I).

We view dialogue as the central interaction concept in coaching. The e-coach uses dialogue for supporting and guiding the coachees following the flow of the CBT coaching process in the effective and efficient completion of the provided exercises. The dialogue not only provides means for information exchange, but also facilitates stages of alignment and negotiation of the therapy plan, and aims to provide supportive motivation for behavior changes. Further, the design and usability of the dialogue generation should take into account the constraints posed by the mobile technology due to their small-size form, such as limited interaction options, restricted memory, power and performance, and unreliable network connection. Heavy and abundant processing still cannot work optimally on the current technology. These issues challenge the extensive natural language interaction between coach and coachee.

To achieve a rich communication between e-coach and coachee as well as to overcome the restrictions presented by the mobile technology, we developed the E-Coach MarkUp Language (ECML). This language specifies the e-coach's dialogue acts at different abstraction levels and forms an interface between the e-coach's communicative abilities and its knowledge base. Before we report on the design and the implementation of ECML, we will present the background of our work in the domain of CBT coaching and dialogue generation. To illustrate the approach, we take examples of dialogue results on a mobile app for CBT-I. Finally, we briefly describe an expert evaluation using scenarios to validate the resulting dialogues.

2. Background

2.1. CBT Coaching Process

CBT is a collaborative and personalized program that helps individuals to identify unhelpful thoughts and behaviors and to learn or relearn healthier skills and habits. It offers a variety of exercise types that differ in aims and properties. For instance in CBT-I, relaxation exercises help a person to relax and to attain a state of calmness; while sleep restriction involves controlling time in bed based upon the coachee's sleep efficiency in order to restore the homeostatic drive to sleep. For monitoring and evaluation purposes, the coachee keeps a diary that will help to identify habits that could contribute to the health problem.

The CBT coaching process is considered as a series of conversations between two individuals - the coach and the coachee[8]. A coach performs persuasive and cooperative behavior to achieve the goals of the therapy; the coachee carries out individual exercises suggested by the coach. There is frequent coordination of activities between coach and coachee. As a cooperative partner, the coach not only helps to set therapy related goals, but also offers support to develop a personal plan and uses persuasive strategies to improve adherence to the exercises. In order to gain the trust of the coachee, the coach should work on at least three requirements[8]: 1. *alignment* about the source of, the importance of and rationale behind exercises, 2. *mutual commitment* between the coach and coachee on the investments in the therapy in terms of closing an agreement, and 3. *adaptation* of the therapy and coaching process to the individual coachee on the basis of the unique circumstances and characteristics of the coachee.

Based on the three requirements, Beun et al.[9] developed the interaction scheme of the coach and coachee (see Figure 1). It consists of a particular order of phases and scheduled activities. The coaching process starts with an introduction phase leading towards a state of alignment and mutual commitment. The next phase involves the coachee performing the actual therapy (by conducting a combination of different CBT techniques). Each technique

is first introduced via dialogues. Then through a series of negotiations, a plan is made for the assignment to which both parties commit. After the coachee has executed the plan for a period of time (often a week), progress and adherence is evaluated. At that point, the coach and coachee can negotiate about a new plan and revise their commitment. The coach monitors and personalizes the coachee's behavior intervention program. At the same time, combined with persuasive strategies, such as adaptation and commitment, the coach encourages the coachee to follow the behavior changing exercises[8,10,11].

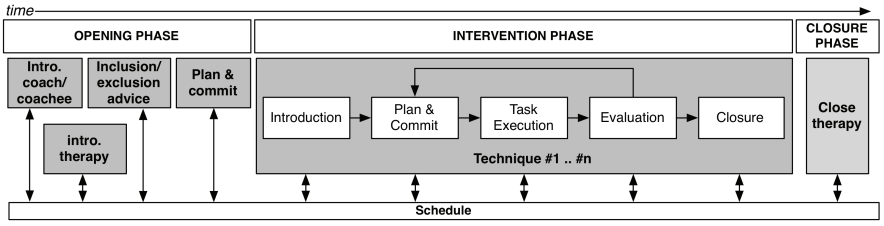

Figure 1 The basic coaching process[9]

2.2. CBT Coaching Dialogue on Mobile Devices

An Eliza-like dialogue system has the capability to conduct conversations with humans using natural language. It works using a limited set of relational templates that is usually represented in a list structure of keywords or word categories and pattern operation rules that exist within the structure. The system uses the structure to maintain the conversation capabilities by substituting a partially composed reply statement template with some portion of the input statement. The transformations to such a structure can be done by a prepared script or scenario that consists of dialogue lines. These dialogue lines are not necessarily sequential; instead, the flow of control among the dialogue lines is guided by the keywords of the input. Recent attempts have been done on increasing the believability of the system, e.g. remembering the previous topic, linking to knowledge base and capturing information during conversation[1,2]. There are other methods on generating dialogue automatically, e.g. using syntax and semantic networks, finite state machines, and so on, however, the template-based approach is still the simplest way. By far the largest amount of work presented by this approach is devoted to develop the large set of pattern-template rules to extend the expressiveness of conversation. Dialogue systems employed in the current behavioral intervention applications combine speech input and embodied conversational agent, and work mainly on stationary devices[6].

In contrast to Eliza where the user determines the topic of the conversation, a CBT e-coach needs control over the topics and the structure of the conversation to maintain the flow of the conversation with regards to the phases and stages of the coaching process. Moreover, to meet the three requirements, i.e. alignment, mutual commitment and adaptation, there must not be any misunderstanding on behalf of the coachee during dialogue. Each dialogue line has to be designed to help the coachee towards the effective and efficient completion of each phase and stage of the coaching process. The topic and content of the conversations correspond to the current state of both the therapy and the individual coachee. This makes that interaction between the e-coach and coachee can be seen as somewhat predictable. Although there should be still some autonomy for the coachee, e.g. to revise the therapy plan, to stop an exercise or to request more information, on each dialogue move, the e-coach has a strict set of dialogue plans and causes the coachee to have only a limited set of options of dialogue discourses. This yields the development of the dialogue generation based on a set of limited responses. Such an approach solves the problem that refers to natural language input on the mobile devices, which is difficult and time-consuming[12]. Moreover, without the natural language processing of input and with the smaller size memory requirement for the limited set of possible dialogue discourses, it is possible for a dialogue system based on the approach to run completely on mobile devices.

Some activities of a CBT program require the coachee to perform them independently from the coach, e.g. sleep restriction. Others require the coachee to use some provided tools (e.g. online diary, relaxation exercise, etc.). These activities require the e-coach to maintain multiple interactions with the coachee over an extended period of time. The interaction frequencies can range from multiple times a day (e.g. for relaxation exercise), to once a day (e.g. fill in diary), or to one or more times per week (e.g. the evaluation of the progress). These interactions are not isolated, stateless sessions (such as in a database question answering system), but require extensive information to be kept

persistent between sessions for a given coachee, with subsequent dialogue tailored based on earlier conversations[1] and all other input sources. Moreover, as some CBT exercises demand extensive self-discipline and stamina of the coachee, intervention is sometimes inevitable, e.g. promoting adherence to the exercises, and requires persuasive and careful interaction. To avoid unnecessary loss in motivation, superfluous interruptions and irrelevant interaction have to be avoided at all cost. The communication should be tailored to the characteristics of the coachee.

We have designed a dialogue system considering the previously mentioned characteristics of CBT coaching and mobile technology. This system runs completely on the mobile device and uses a limited set of response options. In order to personalize the dialogues, the system has direct access to the knowledge base. Information in the knowledge base can be used to tailor a piece of text to the coachee (variables within a text), or to display a completely different text (conditional dialogue actions). The design of the dialogue system is described in the next Section.

3. Design

Figure 2 shows how the e-coach initiates the dialogue with the coachee. A *Conversation* can be triggered by a violation of a constraint or by a scheduled appointment. A constraint relates to a persuasion strategy for sustainable behavioral change[9,13]. Whenever a constraint violation is detected, this violation is repaired by the e-coach by conducting a conversation with the coachee. For instance, when the coachee goes to bed earlier than the agreement, the e-coach resolves this violation by explaining the importance of the assignment. An example of the schedule-based conversations is a periodic evaluation briefing. Since a conversation can be interrupted and all conversations are considered important for a successful CBT program, the e-coach also checks for any missed or unfinished conversation and asks the coachee to continue it before the next conversation starts. All of these conversations are categorized based on their purposes (e.g. introduction or evaluation) and their relation to a certain technique.

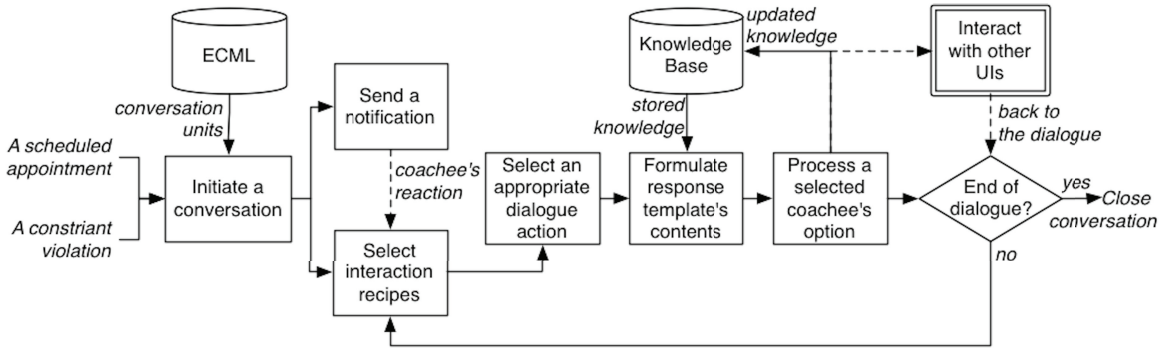

Figure 2 The flowchart of the ECML dialogue generation

The e-coach invites the coachee to a conversation by sending a *Notification*. When the coachee reacts to the notification, the e-coach will play a set of so-called interaction recipes. An *Interaction Recipe* is a playable abstract format that can be called at each stage in the dialogue to prepare the interaction with the coachee based on a script containing a set of dialogue actions. A *Dialogue Action* is a primitive communication act toward the coachee. Interaction recipes group all dialogue actions that represent one e-coach's turn in the conversation and ends with sending a cue to the coachee indicating it is the coachee's turn in the conversation. Interaction recipes also define the flow of the dialogue through the order of the dialogue actions in the script.

Each dialogue action consists of one or more *Templates* that can be selected based on their conditions. The templates themselves provide the contents of the interaction, which renders the communicated message of the e-coach; the conditions define the context in which that particular message is appropriate for the particular coachee. The conditions are expressions referencing concepts from the knowledge base, e.g. sleep diary information in raw ("went to bed at 21:00") or aggregated form ("sleep efficiency was 86% last night"). For example, the interaction recipe for the performance analysis could contain a sequence of dialogue actions that starts with stating the intention

(to discuss the latest progress), then providing the current performance facts and finally giving the performance appraisal. The last dialogue action may contain multiple templates for different ranges of sleep efficiency scores.

A template is defined by the type of the utterance, e.g. statement, question, clarification, etc. Depending on the utterance type, a template can have one or more options for the coachee to respond. Furthermore, both the content and the coachee's options may contain variables corresponding with the concepts in the knowledge base. To produce the dialogue output, the e-coach replaces the variables with their momentary value that it stores alongside the dialogue, because the actual value of knowledge base concepts like "last night" changes over time.

Each *Coachee Option* is linked with one or more actions. An action has several purposes: (i) transferring information directly from the coachee to the e-coach, (ii) allowing the coachee to convey an intention, e.g. to agree with the e-coach, to continue/stop the conversation, etc., and (iii) bridging the dialogue and other input channels. There are five different types of actions:

1. *Update the knowledge base*. For example, updating the value of the phase and stage of the coaching process, the value of the coachee's subjective opinion about the therapy, or the value of the status of a scheduled event.
2. *Go to another activity within the app*. For example, to fill in personal information, to fill in the sleep diary or to adjust the time of a specific appointment in the schedule. After the coachee finishes the activity with the interface, they will be redirected back to the conversation.
3. *Invoke a specific function*. In particular, a function triggers multiple calculations and updates in the knowledge base. For example, during negotiation of the therapy plan, the coachee may reject the e-coach's offer and make a new proposal, which causes the e-coach to re-calculate the offer based on the coachee's proposal.
4. *Execute another interaction recipe*. This action leads to the next state of the dialogue and the selection of an interaction recipe.
5. *Close the conversation*. This action ends the current conversation.

4. Implementation

Following the design in Figure 2, we implemented an XML specification for describing the discourse moves of the e-coach, namely ECML (E-Coach MarkUp Language). Table 1 shows the XML Document Type Definitions of the ECML. It presents the layered, modular structure of ECML. This approach enables the realization of a top layer entity reusing the entities on the lower layers. For instance, a `Conversation` is realized by a set of reusable `InteractionRecipe`s and an `InteractionRecipe` is built up by a script of a sequence of reusable `DialogueAction`s.

Table 1. The ECML - DTD

```
<!ELEMENT ECML (Conversation+)>
<!ELEMENT Conversation (Constraint?, Notification, InteractionRecipe+)>
<!ATTLIST Conversation title #REQUIRED category #REQUIRED>
<!ELEMENT Constraint (#PCDATA)>
<!ELEMENT Notification (#PCDATA)>
<!ATTLIST Notification action #REQUIRED>
<!ELEMENT InteractionRecipe (Script)>
<!ATTLIST InteractionRecipe name #REQUIRED>
<!ELEMENT Script (DialogueAction+)>
<!ELEMENT DialogueAction (Templates)>
<!ATTLIST DialogueAction name #REQUIRED>
<!ELEMENT Templates (Template+)>
<!ELEMENT Template (Condition?, Content, CoacheeOption*)>
<!ATTLIST Template utteranceType (statement|choice|multiple_choice|timepicker|buttons|textfield)
          #REQUIRED>
<!ELEMENT Condition (#PCDATA)>
<!ELEMENT Content (#PCDATA)>
<!ELEMENT CoacheeOption (Text, Action+)>
<!ELEMENT Text (#PCDATA)>
<!ELEMENT Action (parameter+)>
<!ATTLIST Action type (update|execute|tool|function|close_conversation) #REQUIRED>
<!ELEMENT Parameter>
<!ATTLIST Parameter name #REQUIRED value #REQUIRED>
```

The current implemented ECML is written for a six week CBT-I training program. It consists of 20 `Conversation`s. Half of them are scheduled-based and the others are constraint-based conversations. The `Conversation`s use 174 `InteractionRecipe`s that are built up by 260 `DialogueAction`s and 400 `Template`s. To illustrate the dialogue between e-coach and coachee, Table 2 shows two translated dialogue fragments generated during an evaluation of a CBT-I technique: the sleep restriction technique. The e-coach lines are denoted with Ci and the coachee's lines with Uj. The selection of the coachee options is denoted in bold. Such a selection results in different possible dialogue lines of the e-coach, e.g. the dialogue line of C2 varies depending on the answer of U1.

Table 2. Two examples of dialogue in the CBT-I domain. Fragment A illustrates that the coachee finds it difficult to do the assignment and the e-coach tries to motivate the coachee. Fragment B illustrates that the coachee sleeps to little and the e-coach allows her to revise her previous bedtime commitment. The italics show the values that are calculated based on the e-coach's knowledge base. The original text is written in Dutch (see the corresponding screenshots).

	Fragment A
C1:	*Good morning John*. Lets look at the sleep restriction exercise. How did it go last week? Did you find it easy or hard to abide by the new bedtimes?
U1:	[Very easy]; [Easy]; [**Difficult**]; [Very difficult]
C2:	What a pity that you find it difficult. On average, you spend *7* hours in bed, our agreement is *7* hours. You sleep on average *5* hours a night. Although you have difficulty with the exercise, you did adhere to our agreement. You can reward yourself to the sauna.
U2:	[**Continue**]
C3:	Lets have a look at your sleep efficiency. Before you started with sleep restriction: *75%*. Your sleep efficiency last week: *71%*
U3:	[**Continue**]
C4:	You lie awake in bed for quite some time. The best thing for you is to keep these bedtimes for another week. You will probably build up tiredness causing you to sleep longer. Hang in there!
U4:	…

	Fragment B
C1:	*Good morning Emily*. Lets look at the sleep restriction exercise. How did it go last week? Did you find it easy or hard to abide by the new bedtimes?
U1:	[**Very easy**]; [Easy]; [Difficult]; [Very difficult]
C2:	Good for you! On average, you spend *8* hours in bed, our agreement is *5* hours. You sleep on average *3.75* hours a night. You find the exercise easy, yet you spend more time in bed than we agreed upon according to your sleep diary. Let's figure out together why that is the case. What do you think?
U2:	[I don't understand the purpose of this exercise**]**[My bed partner has problems with my new bedtimes] [It's not working with me][**I don't want to talk about this**]
C3:	Lets have a look at your sleep efficiency. Before you started with sleep restriction: *50%*. Your sleep efficiency last week: *54%*.
U3:	[**Continue**]
C4:	You sleep very little to allow you to sleep enough, namely 3,75 hours. To function during the day. I am less strict this week. You can spend *30* minutes longer in bed next week.
U4:	…

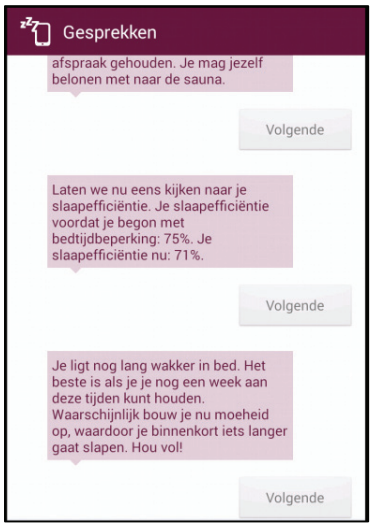

Table 3 shows the scripting specification of the `Conversation` in Table 2. The `DialogueAction`s in the example are used to generate parts of the e-coach's response C2 in dialogue fragments in Table 2. The `Conversation` contains a `Constraint` for ensuring adherence behavior of the coachee during performance of the exercise. This

means that when the e-coach detects that the coachee does not adhere to the exercise (the value of `zeal.SRAdherence.curVal != 1`) before the next scheduled weekly evaluation (`3 < SR.totalDay.sinceConsult < 5`), it will trigger the e-coach to initiate the conversation with the coachee.

During a conversation, the e-coach always tailors its responses based on the current beliefs about the coachee. For example, in C2, the e-coach calculates the average sleep time based on the sleep diary of the coachee. In particular, this dialogue line (C2) is formulated based on an `InteractionRecipe` that contains a `Script` of a sequence of three different `DialogueAction`s. In Table 3, the `DialogueAction`s of the last two sentences in the dialogue line are presented. In this example, the second `DialogueAction` (denoted in bold) contains multiple `Template`s, which can be selected based on their condition. The `Condition` of a `Template` is written as a boolean expression. Each expression in a `Condition` corresponds with a concept in the knowledge base, e.g. the value of the coachee's subjective opinion "difficult" (`feel.SREvaluation.curVal`) variable, is 2 and the value of the coach's objective evaluation "adherence" (`zeal.SREvaluation.curVal`) variable is 1. The use of these variables in the `Content` of a `Template` is distinguished with `${variable-name}`.

Table 3. An example of a constraint-based conversation in ECML. The second `DialogueAction` (in bold) contains multiple `Template`s.

```
<Conversation title="Sleep Restriction Evaluation" category="SR.EVAL.NONADHERENCE">
 <Constraint> sleepDiary.totalEntry.sinceConsult &lt; 3 OR SR.totalDay.sinceConsult &lt; 3
  SR.totalDay.sinceConsult &gt; 5 OR zeal.SRAdherence.curVal == 1</Constraint>
 <Notification action="dialogue">There is a conversation ready about sleep restriction </Notification>
 <InteractionRecipe name="IR_SR_eval1_interrupt#responseSubjEval"><Script>
  <DialogueAction name="DA_SR_eval1#ObjEval"><Templates>
   <Template utteranceType="statement">
    <Content> On average, you spend ${sleep.timeInBed.sinceLastSRConsult.avg} hours in bed, our
     agreement is ${commitments.timeInBed.current} hours. You sleep on average
     ${sleep.sleepTime.sinceLastSRConsult.avg} hours a night.</Content></Template></Templates>
  </DialogueAction>
  <DialogueAction name="DA_SR_eval1#subjObj"><Templates>
   <Template utteranceType="single_choice">
    <Condition>zeal.SREvaluation.curVal == 1 AND feel.SREvaluation.curVal &lt;= 2</Condition>
    <Content>Although you have difficulty with the exercise, you did adhere to our agreement </Content>
    <CoacheeOption><Text>Continue</Text><Action type="execute">
     <Parameter name="target" value="IR_SR_eval1#advice"/></Action></CoacheeOption>
   </Template>
   ...
   <Template utteranceType="multiple_choice">
    <Condition>zeal.SREvaluation.curVal == 0 AND feel.SREvaluation.curVal &gt;= 3</Condition>
    <Content> You find the exercise easy, yet you spend more time in bed than we agreed upon according
      to your sleep diary. Let's figure out together why that is the case. What do you think?</Content>
    <CoacheeOption><Text>I do not understand the exercise</Text>
     <Action type="execute"><Parameter name="target" value="IR_SR_eval1#psycap"/>
     </Action></CoacheeOption>
     ...
   </Template></Templates></DialogueAction>
   ...
 </Script></InteractionRecipe>
...
</Conversation>
```

5. Evaluation

We conducted two expert-based evaluations. The first evaluation was conducted during the development of the CBT-I ECML. It involved two experts in the field of communication and one psychologist specialized in CBT-I. For this evaluation, we collected all possible paths of 14 `Conversation`s and printed them on sets of dialogue tables. We asked the experts to analyze the `Template`s, specifically on their `Content` and `CoacheeOptions`. The evaluation resulted in suggestions for a more meaningful, concise and specific communicated message in `Content`s and more clear and consistent `CoacheeOption`s. We applied the suggestions on all implemented `Conversation`s.

The second evaluation was conducted by running our intervention behavioral app on a smartphone. The same psychologist was involved in this test. For this evaluation, we developed a simulator that could mimic the interaction of a coachee on the app based on a scenario and a predefined list of selected `CoacheeOption`s. The scenario illustrated daily activities of a coachee following a CBT-I session provided by the app. Furthermore, we developed 20 scenarios to represent different coachee behaviors. In 4 scenarios, we described coachees that slept very little

(less than 5 hours a night). In all scenarios, we varied the behavior of the coachee to adhere or not adhere to the agreements, and to show improvement or deterioration in sleep efficiency. These scenarios were used to run and record 20 simulations of the CBT-I training. Table 2 shows two screenshots of the recorded dialogues. We asked the psychologist to analyze the recorded dialogues specifically on the quality of the advice given by the e-coach.

The results of the second evaluation led to minor changes in the communicated message of the `Content` of some `Template`s and rearranging of the sequence of `DialogueAction`s in the `Script`s of some `InteractionRecipe`s. There was no change in the ECML structure. Finally, our sleep expert concluded that the developed dialogue specification was sufficient for an evaluation involving insomnia patients as the end users of our CBT-I app.

6. Conclusion

In this paper, we have reported the development of ECML for scripting discourse moves of a CBT e-coach. The design of the language is based on the basic coaching process developed by Beun et al. [10]. It includes our consideration on the importance of persuasive and careful interaction between e-coach and coachee. We have applied a combination of constraint-based and schedule-based approaches for generating dialogues. The coachee can respond to the e-coach by selecting one of predefined options. These approaches offer fast and efficient computation on a mobile device. In particular, the format of ECML allows a direct access to the e-coach's knowledge base and intelligent processing, which enables the e-coach to tailor the dialogue output to the individual coachee and their circumstances. Furthermore, the specification provides a back-office mechanism for adding and maintaining knowledge of the e-coach. It also bridges the dialogue with other communication channels provided by the app.

The current implemented ECML is written for six weeks of CBT-I training. ECML allows an efficient dialogue specification for the particular therapy that is memory friendly (about 0.5MB). Moreover, the results of our expert evaluations showed that the design had met the requirements of mobile CBT dialogue coaching. Our future work will include an evaluation that involves insomnia patients as the end users of our developed CBT-I e-coach.

Acknowledgements

This research is supported by Philips and Technology Foundation STW, Nationaal Initiatief Hersenen en Cognitie NIHC under the Partnership program Healthy Lifestyle Solutions.

References

1. Bickmore T, Giorgino T. Health dialog systems for patients and consumers. J. of Biomedical Informatics; 2006, **39(5)**: 556–571.
2. Wong W, Thangarajah J, Padgham L. Contextual Question Answering for the Health Domain, J. of the American Society for Information Science and Technology; 2012. **63(11)**: 2313-2327.
3. Bickmore TW, Pfeifer LM, Byron D, Forsythe S, Henault LE, Jack BW, Silliman R, Paasche-Orlow MK. Usability of conversational agents by patients with inadequate health literacy: evidence from two clinical trials. J. of Health Communication; 2010. **15(2)**: 197-210.
4. Dirmaier J, Harter M, Wymann N. A tailored, dialogue-based health communication application for patiens with chronic low back pain: study protocol of a randomised controlled trial. BMC Medical Informatics and Decision Making, 2013, **13(66)**.
5. Wright JA, Phillips BD, Watson BL, Newby PK, Norman GJ, Adams WG. Randomized trial of a family-based, automated, conversational obsity treatment program for underserved populations. J. of Obesity; 2013. **21(9)**:E369-E378.
6. Bickmore TW, Schulman D, Sidner CL. A reusable framework for health counseling dialogue systems based on a behavioral medicine ontology. J. of Biomedical Informatics; 2011, **44(2)**: 183–197.
7. Nguyen H, Masthoff J, Edwards P. Persuasive effects of embodied conversational agent teams. In: Proc. of HCI International 2007: 176–185.
8. Beun RJ. Persuasive strategies in mobile insomnia therapy: alignment, adaptation, and motivational support. Personal and Ubiquitous Computing; 2003, **17(6)**: 1187–1195.
9. Beun RJ, Griffioen-Both F, Ahn RMC, Fitrianie S, Lancee J. Modeling interaction in automated e-coaching: A case from insomnia therapy. In: Proc. of COGNITIVE'14; 2014.
10. Fogg B. Persuasive Technology: Using Computers to Change What We Think and Do. Interactive Technologies. Elsevier Science; 2003.
11. Horsch CHG, Brinkman WP, van Eijk RM, Neerincx MA. Towards the usage of persuasive strategies in a virtual sleep coach. In: Proc. of UKHCI 2012 Workshop on People, Computers and Psychotherapy; 2012.
12. Mackenzie SI, Soukoreff WR. Text entry for mobile computing: Models and methods, theory and practice. Human-Computer Interaction, 17(2 & 3), 147–198, 2002.
13. Ahn RMC. Basic sleep coach knowledge architecture. Technical report, Sleepcare Internal Document, 23112013; 2013.