Building an App for Behavior Change: Case RightOnTime

Piiastiina Tikka, Bereket Woldemicael, and Harri Oinas-Kukkonen

Oulu Advanced Research on Service and Information Systems Research Group (OASIS), University of Oulu, Finland
piiastiina.tikka@oulu.fi, berektka@gmail.com, harri.oinas-kukkonen@oulu.fi

Abstract. The present paper describes and illustrates the process of developing a mobile application intended as a Behaviour Change Support System for people who want to improve their punctuality and time-keeping skills. Design science approach together with Persuasive Systems Design (PSD) model was used. The presented article focuses on the software development process from the perspective of using the PSD model as the central design framework and design science approach (including user evaluation) as the guiding development process.

Keywords: Behavior change; timing strategies; persuasive systems design; design research, systems development, software design.

1 Introduction

Much of the current literature on persuasive systems focuses on describing the behavioral challenge and outcomes after interventions, but the system used in the process is seldom described in clear detail as regards software architecture, leaving developers and researchers alike with limited insights as regards the design of the systems. Such lack of visibility results in so-called black box thinking [1] where it is difficult to assess what the intended behavior change of a system is, and how the software artifact has been constructed in order to support that behavior change objective.

The present paper describes the process of designing and developing a persuasive mobile device app. The design and development work is based on the theoretical frameworks of the Persuasive Systems Design (PSD) model and it employs Design Science Research Method (DSRM) [2].

In this design science research study the selected problem domain for the application is chronic tardiness and challenges with punctuality: the inability to do things in a timely manner. Such issues can contribute to a person's general mental well-being through social pressures and the stress involved; punctuality problems can also affect productivity and success at work or in education [3,4]. The presented software (SW) artifact, a mobile app, provides its users with tools for reflecting upon their actions and behavior as well as a way of practicing time awareness.

2 Background

2.1 Persuasive Systems Design model

According to the Persuasive Systems Design (PSD) model [5] the development of a persuasive system has three main components: understanding key issues of persuasive design (presented as seven postulates), analysis of the Persuasion Context, and the design of system qualities. The seven postulates describe the underlying notions that guide persuasive system development, for example, designers and developers to realize that persuasive systems should always be open, that they should aim at unobtrusiveness, and that persuasion is often incremental [5].

A crucial stage of persuasive systems development is the accurate and careful analysis of the Persuasion Context. The context includes determining what is the intent of persuasion (what the system is expected to achieve), what is the persuasion event, and who is the persuader (also part of the intent of persuasion). The PSD model stipulates that persuasion intent includes the persuader (stakeholders deciding on the system components) and the change type (does the system seek change in behavior, attitude, or both) [5]. The persuasion event in the model accounts for Use Context, User Context and Technology Context. The analysis of the Persuasion Context, then, advises the selection of persuasive design features. The analysis will be presented in section 4.1.

2.2 Behavioral problem domain: punctuality

Up to 20% of people in the USA have issues with punctuality [6]. The reasons for tardiness are various, and such behavior cannot simply be attributed to lack of respect or consideration. The skills to break the bad habits are learnable by practically anyone [3]. A close relation of tardiness, procrastination, can be viewed as a behavior pattern that can lead to negative outcomes such as stress, illness and inferior performance [4]. The social costs of tardiness can also be significant if, for example, a person's career prospects start to suffer because he or she is seen as always late and therefore unreliable [3].

Factors that are common among those people who do not have problems with keeping time include a realistic mindset as regards time, a readiness to plan schedules to tolerate unforeseen delays, a tendency to be organized and to analyze their actions, and not minding having little downtime between activities [6]. Typical tardy personality types include those who enjoy the adrenaline rush brought on by leaving a piece of work until the very last moment before a deadline, and those who enjoy the ego-stroking effect of trying to get as many things done as possible in as little time as possible [3].

These considerations have been taken into account in the Persuasion Context analysis.

3 Method: Design Science Research

The presented development project follows the Design Science Research (DSR), as outlined by Hevner et al. [2]. The research problem in the present paper is to see how to apply the full range of PSD design principles to a specified behavioral problem domain and thus make the role of the information system involved more explicit and transparent, avoiding black box thinking [1]. Additional motivation comes from interest in producing a tool that can help end-users manage time better and learn to be more punctual in their everyday life. The app is available on iOS devices for free through Apple Store.

4 Building a BCSS: RightOnTime app

The software artifact developed to address the identified behavior change targets in the area of time management and punctuality was named RightOnTime in order to suggest timeliness and successful time management. The main features of the app include appointment/event planning ('Activity') and a time awareness practice ('Practice') section where users can train their time-awareness. In addition, the app includes push notification tips and hints that provide the user with suggestions, encouragement and further information regarding punctuality and achieving better time-keeping skills. The tips and hints are intended as support material for the user's behavior change process and support the strategy of direct persuasion. The messages are not tailored to individual users but are general and based on the most common time perception issues identified in the guiding literature [3].

4.1 Persuasion Context analysis

The initial steps of the development process involved the analysis of the Persuasion Context: intent, event, and strategy [1].

Intent. The behavior change intent of the app is to assist end-users to achieve change in their time management and punctuality skills as regards both behavior and possibly also attitude levels [7].

Event. Persuasion event consists of the concepts of Use, User, and Technology Context.

Use Context: The focus of activities in the app is on everyday routines and events – whether private, education or work related – and the content and selection of activities is entirely up to the user. The application encourages planning time use, reducing the larger task of learning punctuality into smaller chunks of individual tasks, and then reflecting on action after set events and activities. These activities stem from the literature [3] suggesting that in many cases tardiness is a result of unrealistic and overly optimistic expectations of time use, and that with better awareness and strategies tardiness problems can be overcome. In terms of SW features, Use Context cate-

gory comprises the overall environment (user interface, data storage, etc.) as dictated by the problem domain.

User Context: The application is intended for individuals who want to develop their time-management skills and become more punctual in their everyday life. The potential users are expected to use the system because they have experienced social discomfort as a result of their inability to be punctual, or have found that better time management skills (and punctuality) would benefit them in their professional (or equivalent) aspirations. Looking at some characteristics of punctual people [3][6] and turning them into practical actions anyone can do, the present project focused on a) learning time-awareness in order to increase the above-mentioned realistic mind-set, b) planning time use, and c) routinely analyzing activities. Each of these user motivations was addressed by supporting increasing awareness and reflection: users were directed to pay attention to their daily appointments and events and then reflect on their own actions [8] afterwards; they were also given means to learn time awareness by measuring how long certain tasks actually take. The tips and hints sent to the users as push notifications were based on existing literature on how to conquer chronic tardiness [3].

Technology Context: The application is developed for mobile iOS devices (namely, iPhone but it also runs on iPads). The benefit of developing an app for mobiles is in making the BCSS available at all times, thereby matching the Use Context as closely as possible. Developing the system for iOS devices necessarily ruled out users of other mobile platforms (Android, Windows, Sailfish, BlackBerry, etc.) at this point in time, but the design principles and persuasive feature set applied in the present project can also be applied to other platforms.

Strategy. The analysis of feasible approaches, namely, planning, reflection and awareness, indicates that the app will use a direct route in the delivery of the persuasive content. Focus on features that require the user to evaluate his or her actions (reflect) promotes the direct route as the user has to elaborate on the actions, that is, engage in intellect based debate over actions and their consequences instead of relying on easily-available heuristics or unfounded assumptions [9]. For example, according to DeLonzor [3] a typical misconception might be that we think that a journey that in reality takes 40 minutes can be done in 30 minutes because there once was an occasion where it happened. Relying on the one instance of 30 minutes uses, essentially, an availability heuristic. By reflecting (elaborating) on the actual time the journey takes as measured and recorded, the system user can re-adjust the time expectation. In addition, overall message quality (including knowing that its source is reliable) promotes use of central processing.

4.2 Persuasive software features

Every section of the application includes a description of its purpose and guidance on how to use it. The app also provides a selection of references to background information on chronic tardiness.

What is more interesting for our purposes here, based on the Persuasion Context analysis, a set of feasible persuasive features was selected from three of the four PSD design feature categories [5]: Primary task support, Dialogue support, and Credibility support. No social support features were implemented. The main features are listed and described in Table 1.

Table 1. Persuasive features implemented in the RightOnTime app.

Primary task support	Reduction	The overall objective of "be more punctual" can seem too big to manage as such, and therefore the objective is split into smaller goals and objectives: individual tasks. Also, by having control over each activity and practice task, the user is allowed to select less critical tasks, thus reducing the possible stress involved if, for example, failure at a task has some serious consequences.
	Self-monitoring	User can do self-monitoring based on self-assessments regarding their own tasks (timeliness and satisfaction). The app also requests user to perform reflection by showing summaries of their daily assessments. In addition, the Practice feature supports observing one's own behavior through set tasks so that it is possible to learn better time estimation.
Computer-human dialog support	Reminders	The app provides reminders for the user-set tasks in itself. The reminders considered here concern the push notifications for users to complete their self-assessments and to view their assessment summaries.
	Suggestion	Tips and Hints push notifications are provided daily (three per day). The content of the notifications provide information, encouragement and guidance as regards punctual behavior.
System credibility support	Real-World Feel	"About" section provides information about the origin, purpose and designers of the app. The development team members' credentials and contact information are included in the "About" section of the app in order to communicate openly who is behind the app and to encourage users to ask for more information if necessary.

4.3 Software architecture

Persuasion Context manifests itself in architecture terms as components for persuasion event (Use Context, User Context and Technology Context) and as a layer of components for persuasion message and route (system-user interaction and system-mediated messages, in the case of RightOnTime) [5][10].

The architecture, seen in Figure 1, shows the four relevant components. The User context involves information about the user, and based on this information appropriate persuasion strategies can be employed. Use Context enables the user interface and data storage to employ the appropriate persuasion strategies. User-system Interaction includes means for the user to set tasks, complete assessments, view results, etc., and it allows the system to record the necessary data about the users and by the users. System-mediated Messages component defines what content and which route is used for conveying behavior change messages to the user. [5][10]

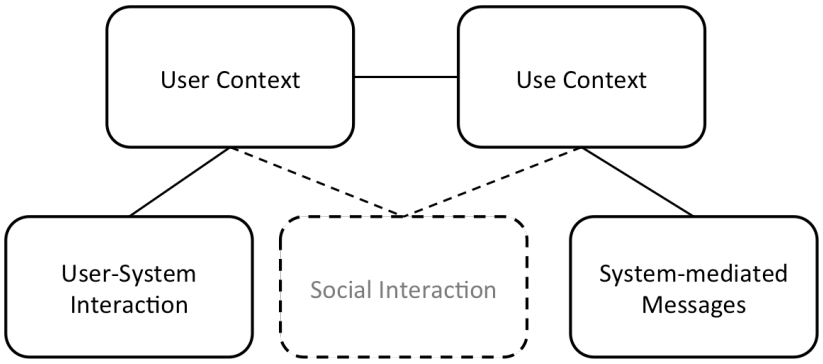

Fig. 1. Persuasion Context software architecture for RightOnTime app, modified from [10]: RightOnTime app does not use Social interaction features.

The technology context (iOS) in the case of RightOnTime has guided the applied software pattern in practice. Apple's application development interface uses Model View Controller (MVC) as a design pattern[1], which involves three groups: the model, the view and the controller. The model handles objects related to data, view handles objects related to user interface and controller is a mediator between view and model object. In our application, model object contains data related to calendar event, local notifications, and user settings. Such data are persistent in the application and have no explicit relationship with the application's view object. The interaction between the application's model and view objects takes place through the controller object. In our application there are three persistent data storages: device's database (SQLite), Calendar and Local notification. A user's setting and user's self-evaluations data are stored in SQLite. iOS application framework provides access to iOS calendar database, and we utilized iOS calendar to store data related to user's events. Data such as notification time and notification message are kept in local notification.

All graphics, the user interface, and interactions are handled by view object. The user can view and update model objects. However, view and model objects do not have any explicit connection and all interaction between view and model object is

[1] Model-View-Controller. Apple Inc. https://developer.apple.com/library/mac/documentation/General/Conceptual/DevPedia-CocoaCore/MVC.html

handled through the controller object. The controller object plays an intermediary role between the application's view object and the model object. It learns updates from a user and notifies the update to the model object. It is also used as a tunnel to display the model object to the user.

The software was implemented in Objective-C programing, which is the primary language for developing iOS and OX software[2]. Objective-C follows syntax of primitive C language, and it has also object-oriented features. In addition to plain objective C-programing language, we have also utilized Apple's foundation framework to develop our application. The app's interface follows iOS human interface guidelines.[3], and the look and feel aims at being "design neutral" in terms of use of any custom components. Instead the look and feel of the app relies on Apple's framework components to enhance the sense of familiarity and trustworthiness that comes with having used the device already.

5 Artifact demonstration

Based on the Persuasion Context analysis, the selected persuasive features (Table 1) were delivered in three main elements of the application: *Activity*, *Practice* and *Tips and Hints*. Visually the app displays a tabular interface with four main sections: Home, Practice, Self-evaluation and Setting sections (Figure 2). Navigation from one section to another is done by selecting a relevant tab button.

Home (view for Activity feature) and Practice tabs: The user can add delete or modify tasks. When there are no tasks on the relevant day, the page shows a brief description of the purpose of the view. The description also includes instructions for how to set new events (helpful for first time use).

Tips and Hints: The tips and hints are shown as push notifications (Figure 3). The timing of these notifications can be random (default) or the user can fix set times for three notification per day under Settings (Figure 4).

[2] Programming with Objective-C. Apple Inc. https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/Introduction/Introduction.html

[3] iOS Human Interface Guideline, Apple Inc. https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/

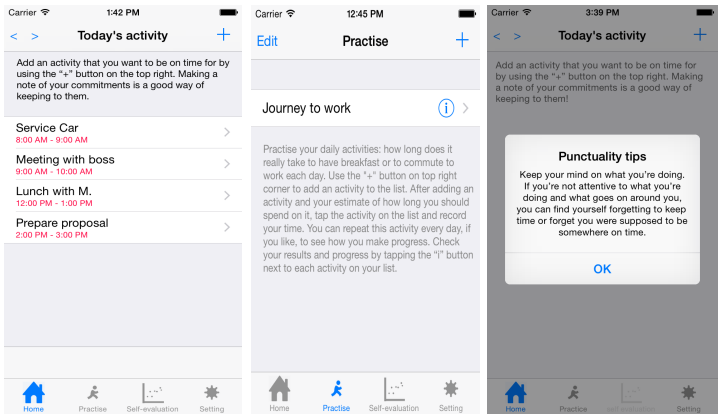

Fig. 2. RightOnTime Home (left) and Practice (center) tabs with example activities, and an example of a tip notification (right).

Self-Evaluation and performance summary views: Upon completion of tasks, the user is prompted to evaluate the timeliness and perceived effort involved in the task, and at a specified time in the day the system prompts the user to review a summary of that day's activity assessments (Figure 3).

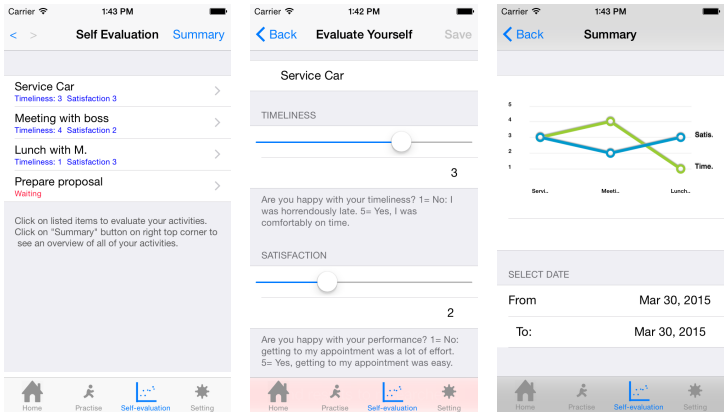

Fig. 3. RightOnTime self-evaluation views: list of activities (left), self-evaluation view (center) and a summary of evaluations of one activity (right).

Setting: Under settings the user can opt to use the system under default settings (random time for all notifications, including requests to provide assessments), or custom settings where user sets times for all notifications and interactions with the system (Figure 4).

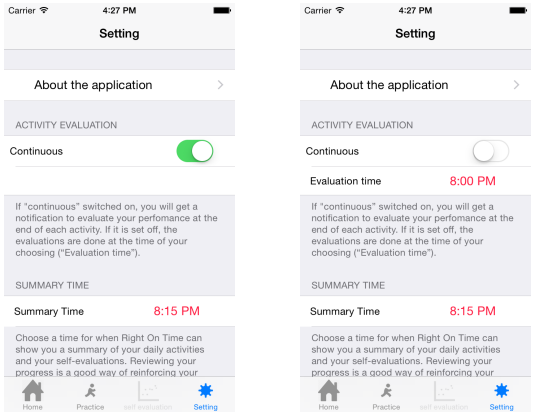

Fig. 4. Settings: Continuous in active mode produces reminders to complete the self-assessments within a 15 minute window after end-time of any user-set task, and to produce three Tips and Hints notifications during the daytime. The randomness is not absolute, but the notifications come during morning, midday and afternoon/evening periods, and not at night. When Continuous mode is in off position, the user can set a time for when he wishes to complete the self-assessments, and when to receive the three Tips and Hints notifications. In each case, summary time can be set by the user.

6 Design science research evaluation

An evaluation of the persuasiveness of the app's main features of Primary task support and Computer-human dialogue support was conducted by means of employing five respondents to use the system for five days. After the experiment period the participants evaluated their experiences on a Likert-type scale and by providing open-ended responses. The questions for the Likert-type ratings were adapted from a perceived persuasiveness questionnaire [11], and these questions were followed by an open-ended text field where participants could elaborate on their rating. All participants responded on an electronic form. The question items were analyzed using a Goal-Question-Metric approach [12]. In addition to the GQM approach, the respondents also completed a desirability test [13] based on an approach developed at Microsoft [14], in which participants were presented with 118 adjectives (60% positive, 40% negative) and they then had to choose five that best described the system. The desirability test is intended to indicate overall satisfaction with a system.

Table 2. GQM items derived from the PSD feature selection (Table 1).

	Goal	Question	Metric
Primary task support	Reduction of larger behavior targets into smaller and more manageable tasks.	Does the application help users to produce target behavior?	Likert /open ended comments.

	The system should help the user understand his or her present time management problems and eventual developments through self-monitoring.	Does the application enable self-monitoring?	
Computer-human dialog support	Remind user of timeliness tasks and of reflection-supporting features (self-monitoring tasks).	Does the app help users to perform required tasks in the application?	Likert /open ended comments
	The system should increase the users knowledge as regards punctual habits and effective time-keeping during the period of system use.	Does the system increase users' knowledge of punctual behavior and how to achieve it?	Likert /open ended comments

Primary task support. Predominantly, the respondents indicated that the app did offer encouragement and support with the primary features of reduction and self-monitoring, thus yielding a **positive response on the Primary task support features**. In the open comments the respondents acknowledged that setting specific tasks helped in seeing time allocation in more concrete terms and to think about what activities need longer or shorter time allocation. Long-term benefits of task setting as a means of observing time use were noted upon, but there was also concern that setting tasks daily for a longer time might become tedious.

Computer-human dialogue support. All but one participant stated that the reminders from the system were helpful in performing the required actions (assessments, reflection). The remaining one expressed a neutral stand on this issue. The division was stronger for the users' assessment on whether they were able to learn meaningful things about timely behavior from the app: all but one agreed (or strongly agreed), where one respondent strongly disagreed. For this respondent the reminder was an irritant and did not have any function in supporting the main task. Therefore, the overall response to the computer-human dialog support features in the app was positive, but the potential of reminders and notifications to irritate users should not be ignored.

Desirability test. Each participant selected five adjectives from a large, pre-set group of 118 adjectives. Of the total 25 selections done by the five respondents in the present evaluation three words were selected by four respondents: *usable*, *personal*, and *clear*. *Innovative* was selected three times; *helpful* and *accessible* were selected twice. *Relevant*, *ordinary*, *effective*, *creative*, *complex*, and *boring* were all selected once.

While *usable*, *personal* and *clear* are positive attributes for the app, as are *helpful*, *accessible*, *relevant*, *effective* and *creative*, we can see that these positive impressions on the use and novelty of the system are also juxtaposed with the minority assessments of *ordinary*, *complex*, and *boring* (design aesthetics elements, largely).

To summarize the results, the features from Primary Task Support category (reduction, self-monitoring, and rehearsal) and Dialog Support (reminders, suggestion) offer encouragement, support and increased knowledge and awareness in the problem domain (that is, higher elaboration [9]). Responses also included concerns over potential irritation that can arise from reminders and notifications, and that continued daily use might become "tedious" for some users. Based on the desirability test and open comments, one area for improvement for the system would be the design aesthetics.

7 Conclusion and Discussion

This paper presented the process of developing a persuasive mobile app based on PSD principles of Persuasion Context analysis leading to selection of persuasive features of the system. The evaluation of the produced app focused on the match between selected features and perceived persuasiveness.

We learned that using the PSD model is a feasible means of defining and describing persuasive software features that match the problem domain. The implications from the development project and this finding are that following a systematic analysis process of the Persuasion Context we can design systems in a way that link intention to outcome in a traceable manner that also allows analysis not only to see whether the system in general affects the outcome, but more precisely which features contribute to the behavior change outcomes and how. Such increased detail in the description of persuasive information systems offers a means of advancing technology development and helps the field in establishing visibility and transparency as regards the technological component in the behavior change process.

As for future research and development of the app, the unobtrusiveness of the system interaction (whether system messages and requests come at a user-defined time or at random times) has been studied [15]. Further research into the effectiveness of this app would have to include a more longitudinal approach with participants who admit to having time management issues and who wish to address these issues in practice. Further development of the app would most likely be based on how the research advises the selection and refinement of the persuasive feature set of the app.

References

1. Oinas-Kukkonen, H.: Behaviour Change Support Systems: A Research Model and Agenda. In: 5th International Conference, PERSUASIVE 2010, Copenhagen, Denmark (2010).
2. Hevner A. R., March S. T., Park, J., Ram, S.: Design science in information systems research. MIS Quarterly, 28(1), 75-105 (2004)

3. DeLonzor, D.: Never be Late Again: 7 Cures for the Punctually Challenged. Post Madison Pub (2003)
4. Tice, D. M., Baumeister, R. F.: Longitudinal study of procrastination, performance, stress, and health: The costs and benefits of dawdling. Psychological science, 454-458 (1997)
5. Oinas-Kukkonen, H., Harjumaa, M.: Persuasive systems design: Key issues, process model, and system features. Communications of the Association for Information Systems, 24(1), 28 (2009)
6. Vozza S.: 4 Habits Of Punctual People. `http://www.fastcompany.com/3025074/work-smart/4-habits-of-punctual-people` (2014)
7. Oinas-Kukkonen, H.: A foundation for the study of behavior change support systems. Personal and Ubiquitous Computing, 17(6), 1223–1235 (2013)
8. Ploderer, B., Reitberger, W., Oinas-Kukkonen, H., van Gemert-Pijnen, J.: Social interaction and reflection for behaviour change. Personal and ubiquitous computing, 18(7), 1667-1676 (2014)
9. Petty, R.E., Cacioppo, J.T.: The Elaboration Likelihood Model of Persuasion. Advances in Experimental Social Psychology, Vol. 19 (1986)
10. Alahäivälä, T., Oinas-Kukkonen, H., Jokelainen, T.: Software Architecture Design for Health BCSSs: Case Onnikka. In: Berkovsky, S. and Freyne, J. (Eds), Persuasive 2013, LNCS 7822, pp. 3-14, 2013. Springer Verlag Berlin Heidelberg.
11. Lehto, T., Oinas-Kukkonen, H.: Explaining and predicting perceived effectiveness and use continuance intention of a behavior change support system for weight loss. Behaviour & Information Technology, 34(2), 176-189 (2015)
12. Basili, V.R., Caldiera, G., Rombach, H. D.: Goal question metric paradigm. In: Encyclopedia of Software Engineering, pp. 528-532. John Wiley & Sons, Inc. (1994)
13. Benedek, J., Miner, T.: Measuring desirability: New methdods for evaluating desirability in usability lab setting. In: Proceedings of Usability Professionals Association, 8-12. (2003)
14. Barnum, C.M., Palmer, L.A.: More than a feeling: Understanding the desirability factor in user experience. In: CHI'10 Extended Abstracts on Human Factors in Computing Systems, 4703-4716 (2010)
15. Tikka, P., Oinas-Kukkonen, H.: RightOnTime: The Role of Timing and Unobtrusiveness in Behavior Change Support Systems. In: A. Meschtscherjakov et al. (Eds.): PERSUASIVE 2016, LNCS 9638, pp. 1–12, (2016)