Journal of Ambient Intelligence and Smart Environments 7 (2015) 37–57
DOI 10.3233/AIS-140299
IOS Press

Cognitive resource-aware unobtrusive service provisioning in ambient intelligence environments

Angel Jimenez-Molina [a,*] and In-Young Ko [b]

[a] *Department of Industrial Engineering, University of Chile, Republica 701, Of. 31, Santiago, Chile*
E-mail: ajimenez@dii.uchile.cl

[b] *Department of Computer Science, KAIST – Korea Advanced Institute of Science and Technology, 291 Daehak-ro, Yuseong-gu, Daejeon, 305-701, Republic of Korea*
E-mail: iko@kaist.ac.kr

Abstract. The cognitive load placed on users by both the proactive and spontaneous provisioning of service functionality and by the physical activities performed in ambient intelligence environments can lead to the depletion of their mental resources. This paper demonstrates how burdening the inappropriate selection of service functionality can be for users by conducting a semi-naturalistic and controlled user test to investigate the significance of the cognitive resource depletion problem in specific ambient intelligence environments. A dynamic service binding and scheduling mechanism is provided based on different types of interference and on mental resources and their demand requirements. A technical evaluation is conducted by simulating the mechanism over a set of various abstract service compositions, making use of real datasets of user interactions with diverse HCI services and daily physical activities. The results show that this mechanism ensures less cognitively taxing, unobtrusive service composition provisioning.

Keywords: Cognitive resource-aware ambient systems, unobtrusive service provisioning, ambient intelligence, service composition, human factors in service computing

1. Introduction

Two decades have passed since Mark Weiser described his vision of users interacting with pervasive technologies "almost at a subconscious level", and one decade has passed since Mahadev Satyanarayanan incorporated the idea of minimal user distraction into the research agenda of ubiquitous computing [56,61]. Many researchers have seen the area of service composition in ambient intelligence (AmI) environments as a natural framework to realize this idea, given its explicit goal of delivering ubiquitous services from a user-centric perspective rather than through systemic channels. Moreover, the literature tends to focus on the architectural infrastructure required by heterogeneous AmI environments to discover, coordinate, and adapt context-aware service functionality based on, among other features, quality-of-service (QoS) requirements and user activities [8,16,24,40]. Nevertheless, there exists the need to assess the cognitive load placed on users by the proactive and spontaneous provisioning of service functionality. This research topic is currently in its infancy [23,32,53].

Service compositions contain services that provide explicit interaction with users [57]. Examples include services that are needed to visualize maps, listen to music or sense vibrations. Users interact with these human-computer interaction (HCI) services using multiple operations, which are called HCI tasks. For instance, the functionality delivered by a document editor on a smartphone requires that users search for

*Corresponding author. E-mail: ajimenez@dii.uchile.cl, Tel.: 56(2) 2978 0543.

options on the screen, push buttons after an option has been found, wait for the option to load, read the information delivered by the option, mentally process such information, determine a path to follow for navigation, push buttons again, and more.

When a user concurrently performs a primary physical activity, such as walking or hurrying on a busy street, and a collection of HCI tasks, they need to allocate mental resources,[1] which divides their attention. Cognition uses mental resources as the main assets to perform decision making, reasoning, perception, etc. [63]. These assets are available in limited quantities in human information processing systems, which suppose a limit on the joint demand of cognitive resources [46]. The core mental resources assigned during information processing are attention, perception, long-term memory, working memory, and motor control [27,49]. Therefore, both the HCI tasks and the user's primary physical activity compete for the same amount of limited mental resources.

The side effects of this competition on the user's performance are termed as cognitive resource depletion (CRD) in the area of cognitive psychology [63]. This leads to distractions; increases errors, stress, and frustration; and reduces the ability to perform mental planning, problem solving, and decision making [20,42,45]. In particular, recent studies have shown how annoying technostress is to users [38,39,48]. A naturalistic and controlled user test is conducted to demonstrate how burdening the inappropriate selection of service functionality can be for users in AmI environments. This study demonstrates the relevance of the CRD problem in such environments.

The emotional pressure caused by these stressful conditions hampers the user's engagement and mental resources, as shown by a growing body of knowledge that highlights the interdependence between emotion and cognition [52,58], decreasing the ability of a user to perform physical activities and HCI tasks. Because this behavioral effectiveness is governed by a set of cognitive processes, the above problem can be avoided by delivering unobtrusive service functionality. Specifically, services that aggregate cognitive demands, together with the cognitive demands of physical activities, should not overwhelm the user.

Such a challenging issue is addressed in this paper by a novel cognitive ergonomics mechanism for identifying service functionality during runtime in accordance with situational demands of mental resources by user's activities and HCI tasks. The proposed approach is based on two theories from cognitive psychology: the human processing system theory of Navon et al. [46] and the multiple-resource theory (MRT) of Wickens [63]. The former is based on the idea of a limited amount of underlying resources available at any moment in the human processing system [44]. These mental resources are demanded at different degrees by time-sharing activities during any given situation. In contrast, the MRT highlights the competition and interference among mental resources allocated in different physical/cognitive structures. It is based on a practical description of the different mental resources required by activities. According to this model, the performance of time-shared activities is sensitive to their combined difficulty and overlapping of common mental resources.

The major technical contributions of this paper are twofold. First, a service selection mechanism that utilizes a cognitive-resource-aware description of physical activities and HCI tasks is proposed. Second, this model is leveraged to dynamically bind and schedule abstract service components to concrete services by considering different types of interference among mental resources and mental resource demands either in sequential or concurrent service interaction behaviors. A technical evaluation is conducted by simulating the dynamic binding and scheduling mechanism over a set of various abstract service components, making use of real datasets of user interactions with diverse HCI services and daily physical activities. The results demonstrate that the mechanism is effective in terms of finding appropriate HCI services to instantiate time-shared abstract services and physical activities. Moreover, the mechanism scales well in a mobile setting.

This paper is organized as follows. Section 2 describes and analyzes work related to this research. Section 3 provides the required background on the CRD problem and the psychology concepts utilized in this research. This background is explained through an example scenario. In addition, this section shows how to compute the degrees of interference and cognitive resource demands. Section 4 consists of a semi-naturalistic, controlled user study with real users that is used to assess the relevance of the CRD problem in AmI environments. Section 5 introduces the cognitive-resource-aware dynamic service binding and scheduling mechanism utilized to identify concrete HCI services in light of concurrent service functionality and physical activities. Section 6 shows the experimental

[1] "Mental", or "cognitive", resources are used interchangeably throughout this article.

results that were obtained by simulating the mechanism on real datasets of user interactions with service functionality. Section 7 discusses the major findings, contributions to the state of the art and disadvantages of the approach. Finally, Section 8 concludes the paper.

2. Related work

The following research fields are meaningful to this work. *Multimodal interfaces* (MUIs) concern delivering HCI functionality without cognitive conflict at the user interface level. *Interruption management* specializes in delivering functionality with the correct timing to avoid interruptions, and the last area concerns the *optimal, dynamic selection of services* for instantiating abstract compositions. To the best of our knowledge, this approach is the first to integrate the benefits of all of these areas.

2.1. Multimodal interfaces

The goal of MUIs is to reduce the user's effort in terms of perceiving their effects in different interactive contexts. This approach makes use of the user-centred design to assess the interactivity between users and ubiquitous services from a perspective of users' goals [29,37,60]. MUIs are based on multiple input streams, which require parallel cognitive processing. This is risky in the sense of leading to interference among the attributes of mental resources. MUIs enable users to enhance their perceptual and verbal response capabilities during an interaction with an interface [21]. MUIs, such as speech, pen, touch, and gesture interfaces, are meaningful in the domain of HCI in terms of optimizing the delivery of service functionality to avoid cognitive interference among the modality attributes (auditory, visual, or touch) of mental resources. As shown in [50], over 95% of users are shown to prefer multimodal interactions over unimodal interactions. This study also shows that a user-centric design for an MUI helps to decrease the cognitive load on users [51]. Kong et al. [34] proposes a human-centric approach for an adaptive MUI that consists of mapping a modality space to a user's preferences space. The major drawback of this approach is the need to define the interaction scenario in advance.

Nevertheless, all of these approaches do not address the challenge of adaptively delivering an MUI in accordance with both user behavior while interacting with HCI tasks and the physical activity in which the user is involved. They assess, in an exclusive manner, either the mental resources derived from the user's activity, from computing interfaces, or from QoS requirements defined in advance.

In contrast, in this paper the mental resources from all of these sources are assessed, considering not only the modality attribute but also other types of attributes, as described in Section 3.

2.2. Interruption management and job design

Interruption management is one of the major challenges in reducing mental workloads [26]. Although application notifications facilitate user tasks, they can significantly lower performance in terms of the user's ongoing task [6,19,22,35,36,54]. The burden of notification is significantly smaller when the services are delivered to a user whose mental workload is light. This is why the goal of interruption management systems is to manipulate the notification delivery time to decrease the cost of mental interruption. As shown in [12], a notification while the user is engaged in a primary task can reduce task performance by 30%. The same author shows in [11] that the cost of interruption can be lowered by delivering the notification at moments of lower mental workload. These moments occur at subtask boundaries or during task switching. However, these systems can only react using the predesigned association between a notification and the user task. In addition, they do not include multiple time-shared services.

The notification platform system shown in [19] is an interruption management system that allows users to tailor their costs of interruptions. The drawback to this approach is the static association between situations and software. Therefore, the major drawback is reactivity. Specifically, notification platform systems can only react in the manner that the strategy has defined in advance. In addition, the system does not consider what service functionality to deliver to decrease the mental workload; rather, it considers when to deliver it. Using the dynamic service binding and scheduling mechanism, this paper focuses on both aspects, as shown in Section 5.

Job design approaches are popular in industrial domains such as aviation, manufacturing, and automotive industries [44,45]. Their goal is to set an optimal cognitive load for software to reduce the mental effort expended by operators of process control systems. Such studies utilize a cognitive task analysis based on a theory of cognitive task load that includes (1) percentage

Table 1
Comparative analysis of the related work

Approach	Activity-aware	Cognitive Resources Modeling	Dynamicity	Metrics to Assess CRD	Mobile AmI Environment
MUI Systems	*Limited.* A dimension of the interaction context	*No.* Only the modality attribute	*Yes.* Interaction context aware	*No*. Focused on decreasing users' cognitive load	*Limited.* Preferably MUI in desktop environments
Interruption Management	*Limited.* Subtask boundaries, task switching	*No.* Users tailor their cost of interruption, or reason about moments of lower mental workload	*No.* Only react in the way the strategy was predefined in advance	*No.* Only tries to decrease the cost of mental interruption	*Yes.* Specially to manage notifications to the user
Job Design in industry	*Limited.* Makes a decomposition of the activity into the percentage of time occupied, the level of information processing, and the number of task-set switches	*No.* Only an estimation of the level of information processing	*No.* Requires cognitive task analysis in design time. Do not adapt the application during runtime	*Limited.* Multidimensional on the percentage of time occupied, the level of information processing, and the number of task-set switches	*No.* Naval ship control centre, avionics, in-vehicle information systems (driving safety)
Service binding and reconfiguration	*Yes.* Context aware QoS based service composition approaches	*No.* This is our contribution	*Yes.* Binding and re-binding of abstract service compositions	*No.* This is our contribution	*Yes.* But mainly focused on Web services in general

of time occupied in the operation, (2) level of information processing required by the operation, and (3) the number of task-set switches performed while engaged in the operation. The major drawback is that this model has only been shown to be successful for long-term tasks with very stable task-set switch frequencies. There is no evidence of its practicality in short-term, multiple-interaction HCI tasks.

2.3. Dynamic service binding and reconfiguration

This is an important category of study for the requirements of dynamically binding and scheduling services. The literature has tended to focus on the instantiation of abstract service compositions at runtime. The selection of the service is performed using an optimization problem based on various parameters such as the Quality of Service (QoS) [41]. Because user behavior, the environmental context, and services in AmI environments are highly dynamic, services must be continuously monitored and reconfigured [17].

The approach of this paper has partially been inspired, although in a different domain, by the influential studies of Ardagna & Pernici [9] and Alrifai & Risse [7]. The former work ensures the optimality of the service components by periodically performing additional reconfigurations. The reconfiguration period is adjusted throughout the service process based on deviations from the environmental context. The latter proposes a heuristic to find close-to-optimal service compositions.

Despite efforts to address these challenges, to the best of our knowledge, this paper is the first attempt to propose the dimension of cognitive resources as a first-order class to ensure human-processable service functionality.

Table 1 summarizes the span approaches in the literature review.

3. Background

The following simple scenario shows the technical challenges that need to be addressed to avoid CRD while time-sharing a primary physical activity and HCI tasks:

Scenario Juan is rushing to a meeting downtown. He listens to music while walking through the busy streets trying to find the restaurant. He realizes that he is close to the meeting point but cannot determine the exact location of the restaurant. He decides to interrupt his physical activity, stepping aside on the street, takes out his smartphone, launches a map application, and types in the address of the restaurant. After waiting too long for the address, he finally arrives at the meeting; unfortunately, he is late. Because he was listening to music

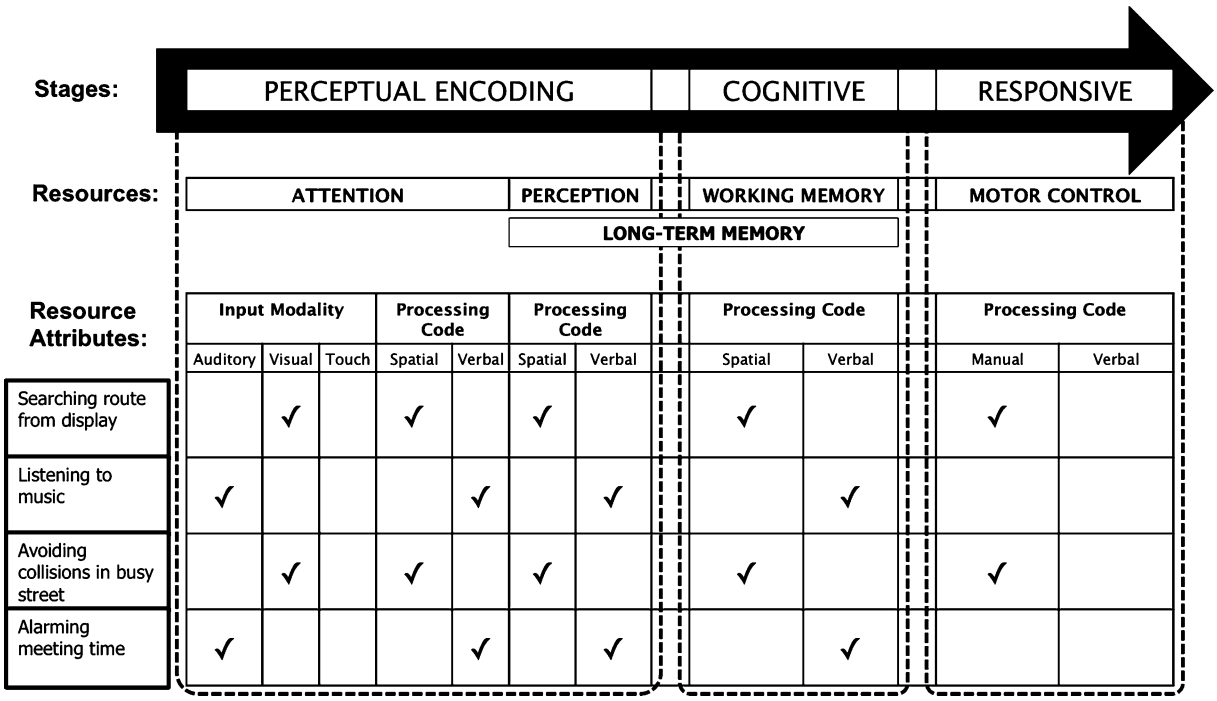

Fig. 1. The human processing system, the multiple resource competition framework (adapted from [63]), and an illustrative scenario sample.

and looking for directions, he missed the sound of the alarm notifying him of the meeting time.

Pablo is going to meet Juan. He also needs directions, but he verbally accesses the map application. This is launched as the media player volume is lowered, and a voice recognition application directly inputs Pablo's query. Pablo receives audio directions to the restaurant as he walks and listens to his favorite song. Pablo notices the alarm and runs to the location three minutes prior to the appointment time. The alarm application had switched to vibration mode because Pablo was receiving a considerable amount of auditory stimuli.

Juan and Pablo arrived to the restaurant, but both solutions are very different in terms of their users' experiences. Juan had to interrupt his primary activity to manually perform different HCI tasks, thus wasting time. In contrast, Pablo never interrupted his primary activity and was smoothly aided to the restaurant, thus arriving on time.

Based on the MRT, it is clear that the CRD that may arise from certain configurations of time-shared HCI tasks and a user's activities can lead to a decrement of user performance in AmI environments. Moreover, in the 1980s, Wickens found empirical evidence of three cognitive dimensions that may lead to cognitive interference when they are shared by multiple mental resources [63]. This interference results in different levels of time-sharing efficiency during competition for the available mental resources. For instance, Juan tries to see the map application while attempting to avoid collisions on the sidewalk. Both activities use vision (focal and ambient), a non-sharable input modality.

As shown in Fig. 1, the first dimension consists of three processing stages: *perceptual*, *cognitive* or *central processing*, and *responsive*. Evidence from the field of psychology shows that perceptual and cognitive stages use common mental resources – *sensation* to intake external stimuli, *attention* to arrange sensed information, *central executive* for abstract control of cognition, and *working memory* to retain short-term information, among other resources. The simultaneous demand of perceptual, cognitive, or both perceptual and cognitive dimensions of mental resources produces processing stage interference. For instance, as exemplified in Fig. 1, the HCI task of understanding the trajectory of the route provided by the map application (cognitive stage) interferes with the HCI task of understanding the lyrics of the song that Juan is listening to (also cognitive stage). Both require working memory. In contrast, the responsive stage uses a separated set of resources, such as *motor control*, for walking or typing on tiny screens.

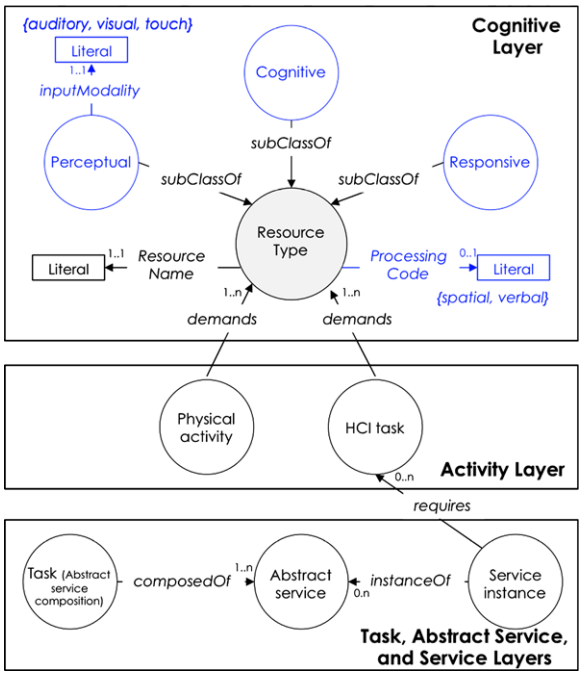

Fig. 2. Cognitive resource aware activity and service description model.

The second dimension consists of the perceptual modalities of mental resources. These modalities explain why the time-sharing performance of activities is better when sensing two stimuli with *visual* and *auditory input modalities* – i.e., cross-modal time sharing. In addition, they explain why the intra-modal intake of stimuli negatively affects performance, producing input modality interference such as auditory-auditory (when Juan misses the sound of the alarm notifying him that the meeting time is approaching while he listens a song), visual-visual, or tactile-tactile interference. In contrast, processes operating in different codes – either *verbal* or *manual/spatial* and regardless of their processing stages – utilize mental resources that are structurally separated. In this sense, processing code interference will arise if a user simultaneously tries to drive and manually dial their smartphone. Both the primary physical activity of driving and the HCI task of dialing require a manual processing code.

The major aspects of the MRT and the human system of information processing have been embedded into a cognitive-resource-aware activity and service description model, which is described in a previous work [30]. This description enriches the service profile by adding a cognitive layer and is shown in Fig. 2.

The core elements of the model consist of the cognitive resource type, the physical activity, the HCI task, and the service functionality. The input modalities, processing stages, and processing codes are added as properties of the cognitive-resource-type object. In contrast, the delivery of service functionality may require HCI tasks and in turn cognitive resources characterized for these properties.

3.1. Computing the degree of cognitive resource interferences

The cognitive interference found by Wickens can be explained in the framework of the domain of this paper as follows:

- *Input Modality Interference*: This occurs when resources from the physical activity and the HCI task have the same input modality. If this occurs, the perceptual dimension stops processing, thus interrupting the HCI task. For each input modality, this interference value is set to *complete* if the physical activities and HCI tasks simultaneously present the same input modality. They are set to *none* otherwise.
- *Processing Stage Interference*: This accounts for interference involving the three processing stages. Intra-stage interference is only produced at one processing stage, while cross-stage interference simultaneously involves perceptual and cognitive stages. The degree of interference for (1) perceptual stage and cognitive stage interference can be described as *complete*, *partial* or *none*, while for (2) the response stage, the interference can be described as *complete* or *none*. For each process-stage interference p in (1), the value for p is *complete* if the physical activities and HCI tasks simultaneously demand the same processing stage dimension. This is set to *none* otherwise. The value for each processing-stage interference q in (2) is *complete* if the sum of the demands of the physical activity and HCI tasks for cognitive resources exceeds the processing capacity of q, *partial* if it does not, and *none* otherwise.
- *Processing Code Interference*: Intra-stage code interference involves HCI tasks and physical activities using cognitive resources with the same processing code operating at the same processing stage, while for cross-stage code interference, the cognitive resources have the same processing code at two different processing stages. For each processing code, the interference value is *complete* if the physical activities and HCI tasks us-

ing cognitive resources have the same processing code. They are *none* otherwise.

3.2. Computing the degree of cognitive resources demand

The performance of physical activities and/or HCI tasks by a given user is directly proportional to the availability of their mental resources to the extent that the depletion of these resources limits the realization of the activity [46].

Hence, a service composition should consider the availability of mental resources to ensure that its deployment will be beneficial to the user and not overwhelming. This can be ensured by verifying that the total amount of mental resources demanded by physical activities and HCI tasks are less than the user's total processing capacity.

The existing literature [46] assumes that this threshold is a fixed amount and that it can also be calculated empirically for highly demanding situations such as hurrying through a busy street. Metrics to assess the demand of mental resources range from subjective measures – e.g., the SWAT scale and the NASA TLX scale [55] – to physiological measures – e.g., heart-rate variability, pupil dilation, and visual scanning [25]. This is consistent with the idea of a limited human central processor and also gives an upper bound for the determination of feasible HCI tasks to be concurrently performed with a given physical activity. More details are given in Section 5.

4. Relevance of the problem: Attention allocation analysis

A semi-naturalistic and controlled user test is used to investigate the significance of the CRD problem in AmI environments. Specifically, this user test attempts to demonstrate how burdening the inappropriate selection of service functionality can be for real users in such environments.

Twenty Engineering students (age range = 18–28, gender: 30% female and 70% male) are used in two test cases composed of time-shared physical activities and HCI services deployed on a smartphone. The first group (the baseline) consists of highly cognitively demanding HCI services and activities as well as cognitively interfering activities and HCI services. In contrast, the second group is designed to not include interference and to exhibit low cognitive demands. The following is an example of test cases in the first group:

- The participant has a conversation at a coffee shop with the experimenter on a topic of interest to the participant in an effort to encourage her to express own opinions. Simultaneously, the participant listens to music with lyrics in her mother tongue from headphones plugged into the smartphone (only one earphone was used) while she drinks a cup of coffee provided by the experimenter.
- The participant has to search for specific books indicated by the experimenter from the bookshelf of the campus library. The participant listens to music in the same manner.

While time sharing with these activities, the participant interacts with the following sequence of HCI segments composed of HCI tasks, which are performed using a smartphone:

- Reading an instruction to watch a video
- Watching a video
- Tagging the video (selecting existing tags and writing new ones)
- Writing comments about the video
- Reading instructions about selecting friends to share the video with
- Selecting friends from a list

In contrast, for the second group of test cases, a participant performs the above HCI tasks while engaged in the following activities:

- The participant in the same coffee shop only drinks but neither listens to music nor participates in a conversation.
- The participant seated at a laboratory desk simply controls her space.

4.1. Methodology

Eye movement measures are conducted, widely used in the HCI area, to elicit distractions in dual or multi-task interactions. Specifically, it is observed how users divide their attention to perform physical activities while simultaneously interacting with spontaneously delivered service functionality compositions. The rationale is that attentional processes are what govern the allocation of mental resources in the human information processing system. Moreover, techniques that monitor peripheral signifiers exposed to the environment, such as a visual gaze, are among the few techniques that can be used to assess the human information processing system [53]. On the other hand,

the justification for monitoring the user's visual gaze is that holding the user's attention improves the performance of both the HCI tasks of an HCI service [49,53] and the mobility activities in a given situation [10,22,44,45,63]. Specifically, the adequate allocation of attention is an important enabler that determines the success of a user-centric task. The objective is to assess the effects of CRD on the user's performance caused by inappropriate configurations of HCI services and activities.

The following characteristics of the user's visual gaze are monitored: (1) the duration of continuous attention to the smartphone at each segment of interaction (HCI segment), which is measured in seconds; (2) the number of attention-switches to the environment at each HCI segment, which is measured as a frequency; and (3) the duration of attention-switches before returning to the HCI segment on the smartphone, which is measured in seconds.

The first metric enables an estimation of how much time mental resources are allocated to intake the service functionality, to process it in the central executive stage, and/or to rehearse and execute an adequate response either vocally or manually/spatially. Long durations of continuous attention to HCI segments on a smartphone constitute an estimated assessment that the user is not interrupted by activity demands. In addition, this is evidence that the limited capacity of the processing system is not affected by information overload coming from both the HCI segments and/or the activities.

The second metric estimates the interference intensity among mental resources demanded by activities and HCI segments. Specifically, a high number of switch-off instances from the smartphone would indicate that the same mental resources allocated to the HCI segments are required to satisfy the execution of the activities.

Finally, the third metric assesses the importance the user gives to the activities, overriding the HCI segments. Long attention-switching durations before returning to the smartphone indicate that the user's attention is captured by the activities. This may due to the cognitive demand of the activities or due to other external factors such as the user's preferences or interests.

4.2. Procedure

To gather data, it is used a mini-camera attached to a holder, which video tapes the fingers' interaction with the smartphone, as shown in Fig. 3a. This camera is located at an appropriate distance above the smartphone screen to ensure an angle that completely covers the smartphone screen. The visual gaze is videotaped by a mini-camera held by the experimenter placed in situ to shadow the user.

At the beginning of each test case, the participant was asked to read and sign an agreement of participation. This document stated that the information gathered from the user study would be used anonymously. It also stated that the personal behavior inferred from the study would be kept private. This document also contained a section containing personal data such as age, gender, occupation, and experience with smartphones or cellphones. The participant was then trained for approximately ten minutes with a small set of HCI tasks. Most of the participants already had experience with such interactions on an Android smartphone; however, this training was strictly repeated for all the users. This training was performed for the first test case in which the participant was involved in the study. Different participants started with different test cases to avoid the effect of automaticity on interactions with the smartphone.

After the completion of the training upon the first scene in which the participant was involved, she/he was informed that a set of spontaneously delivered "applications" and their respective instructions would appear on the screen. She/he simply needed to adhere to the instructions and perform the indicated steps as well as possible while simultaneously completing the "physical activity" in which she/he was supposed to be involved. The terms "applications" and "physical activities" are used in a colloquial manner to avoid jargon and unnecessary technicalities. The circuit of the test cases varied amongst different participants. After the completion of one test case, the participant, the experimenter, the assistant user, and the apparatus were moved to the location of the next test case. The complete circuit took each participant approximately three hours to complete. After the completion of all of the test cases, participants were informed of the purpose of the user study but were asked not share the content and purpose of it with anyone else (especially with people they might know who would also participate in the future).

4.3. Data analysis and results

A set of 80 movie files captures the interactions with the smartphone and the visual gazes of the users

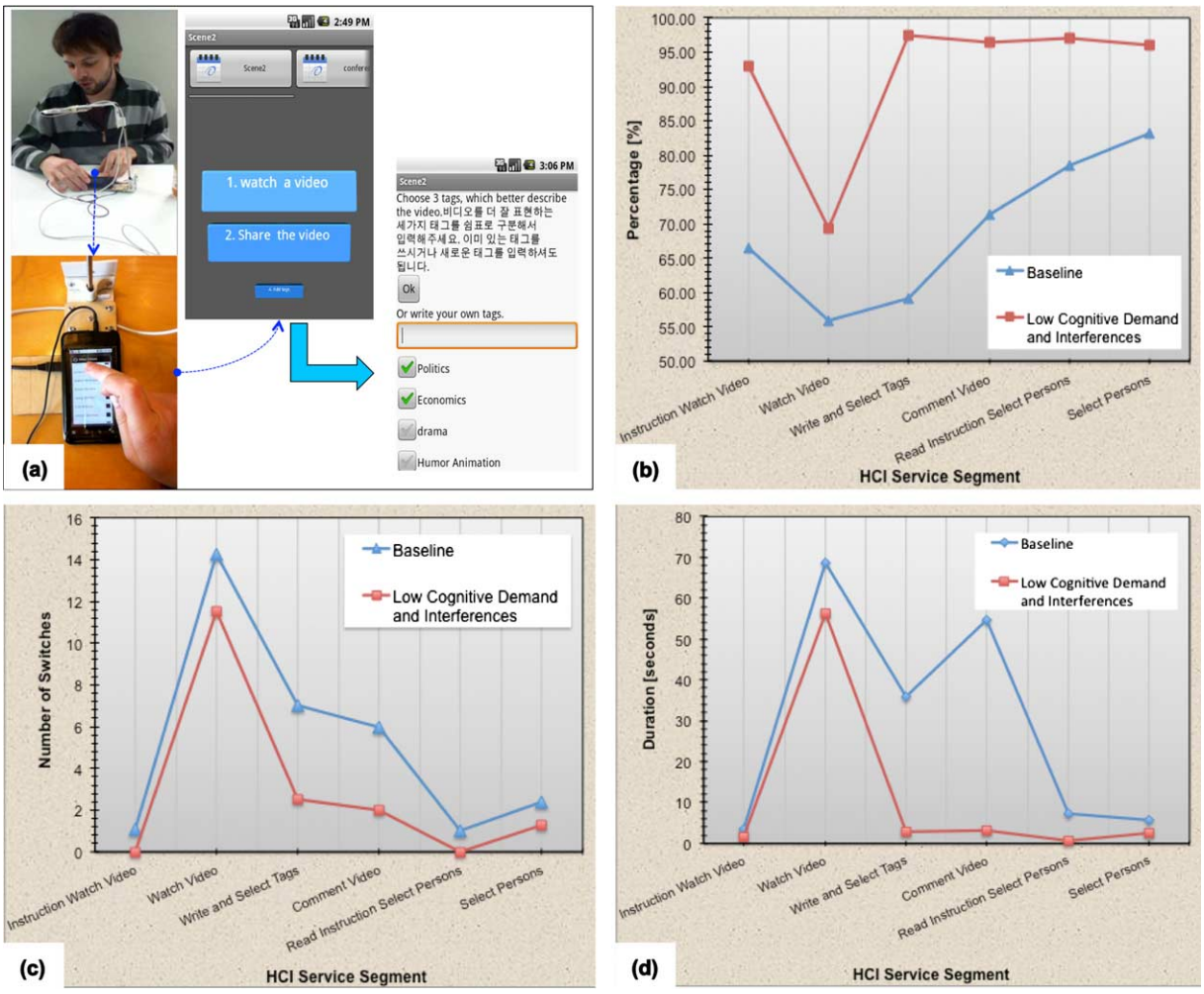

Fig. 3. (a) Apparatus. The results of the attention allocation Analysis: (b) the percentage of continuous attention given to the smartphone at each HCI segment is higher for the group of non-cognitively demanding and interfering test cases (second group), (c) the average number of attention-switches to the environment at each segment is lower for the second group, (d) the duration of the attention-switches before returning to the HCI segment at the smartphone is also lower for the second group.

throughout each test case. From these movie files, the participants' behaviors for 480 HCI segments were extracted. Each segment is carefully analyzed by visual inspection to gather the data necessary to obtain the values of the three metrics. The following data was obtained: (1) the start time of each HCI segment; (2) the end time of each HCI segment; (3) the exact time the participant begins their visual interaction with the smartphone after an HCI segment has started (the value of which may not necessarily coincide with the starting time of the HCI segment); (4) the start time of each attention-switch from the smartphone to the environment, and (5) the end time of each attention-switch from the smartphone to the environment. The precision level was ensured based on the minute:second:frame format.

Figure 3b shows the average amount of time (in percent) participants attend to the smartphone at each HCI segment for both the baseline (mean = 69.1%, min = 55.89%, max = 83.08%, SD = 10.6%) and the non-cognitively demanding and non-interfering second group of test cases (mean = 91.5%, min = 69.3%, max = 97.54%, SD = 11%). It is clear that the participants tend to allocate a greater amount of their attention to the smartphone in the second group. The improvement in a less cognitively demanding HCI service segment such as `Select Persons` is approximately 15%; however, in a highly demanding segment

such as `Write and Select Tags`, the difference can be as high as 46%. The average improvement can be as high as 22.4%.

Note that in the baseline, participants interact with a music player HCI service. This HCI service is performed while the other HCI service segments are performed. Moreover, the listening HCI task required by the music-playing HCI service strongly demands selective attention, whose input modality is auditory. The processing code of this selective attention resource is verbal. In addition, this HCI task demands, to a lesser extent, a perceptual mental resource, whose processing code is also verbal. Both mental resources, selective attention and perception, belong to a perceptual processing stage. Finally, this HCI task also strongly demands the central executive processing – also known as working memory – mental resource. The processing code of the working memory for this case is verbal, and its processing stage is cognitive. Thus, because having a conversation demands the same type of mental resources, there is competition for these resources that justifies the degradation of the continuous attention to the smartphone.

The same applies to the mental resource interference. There was a verbal-type processing code interference between the user's activity – having a conversation – and the HCI tasks of writing and selecting tags, reading instructions, watching a video, commenting on a video, and listening to music.

This processing code conflict affected both the central executive processing stage and the responsive stage. The justification for this is that to rehearse a response to the interlocutor in a conversation and simultaneously understand the content of HCI segments – such as instructions to follow, reading short texts, searching for specific peoples' names in a display, or understanding the lyrics of songs – the participant needed to process information in a vocal dimension with the central executive system using working memory as well as long-term memory.

The average improvement in the number of attention switches to the environment was 59.5%. The duration of attention switches before returning to the HCI segment saw an average improvement of 67.2% (see Figs 3c and 3d).

The question that arises is how to select the appropriate service functionality for a user without cognitively burdening the user in terms of the joint performance of HCI tasks and a primary physical activity.

5. Dynamic (re)binding and scheduling: Interleaving user interactions and service selections

Initially, it is assumed, based on the state of the art of context-aware activity recognition in AmI environments, that the primary physical activity with which the user is engaged can be recognized during runtime and with high precision. This has been realized by ubiquitous intelligence, which aggregates increasingly rich sensing information extracted from public spaces, space semantics, and personal schedules, among other types of context information [31,33]. These technologies are making it easier to spontaneously configure and deliver service compositions to users in AmI environments.

Second, it is assumed that service compositions that are represented in the BPEL (Business Process Execution Language) and that are stored in a service composition repository consist of sequential and parallel constructs of abstract services coordinated by control-flow patterns (AND-Split/Join, XOR-Split/Join, OR-Split/Join, etc.) [62]. Service instance functionality is described in WSDL (Web services description language) documents, which are stored in a UDDI (Universal Description Discovery and Integration) registry. In addition, it is assumed that each service instance belongs to a list of functional equivalent services representing a specific abstract service type. This equivalence applies to both the functionality and the QoS, which are represented in the interface of each service instance.

Some existing concepts were extended into the service engineering area to apply the cognitive-resource-aware approach to the service selection problem. The first distinction refers to implicitly/explicitly interacting services [57]. The former consists of services running in the background. The latter are services with which the user interacts (HCI services) by performing a set of HCI tasks.

Each HCI task is associated with a cognitive dimension of service (CoS) profile, which provides a description of its cognitive dimensions, as determined by service developers through easy-to-conduct task analyses.

Another important distinction is that of service concurrency. This refers to a collection of time-shared abstract services from different branches of a parallel construct. The task selector stochastically estimates the most probable time window of concurrent execution for these abstract services based on the probability density functions of the user's interaction time

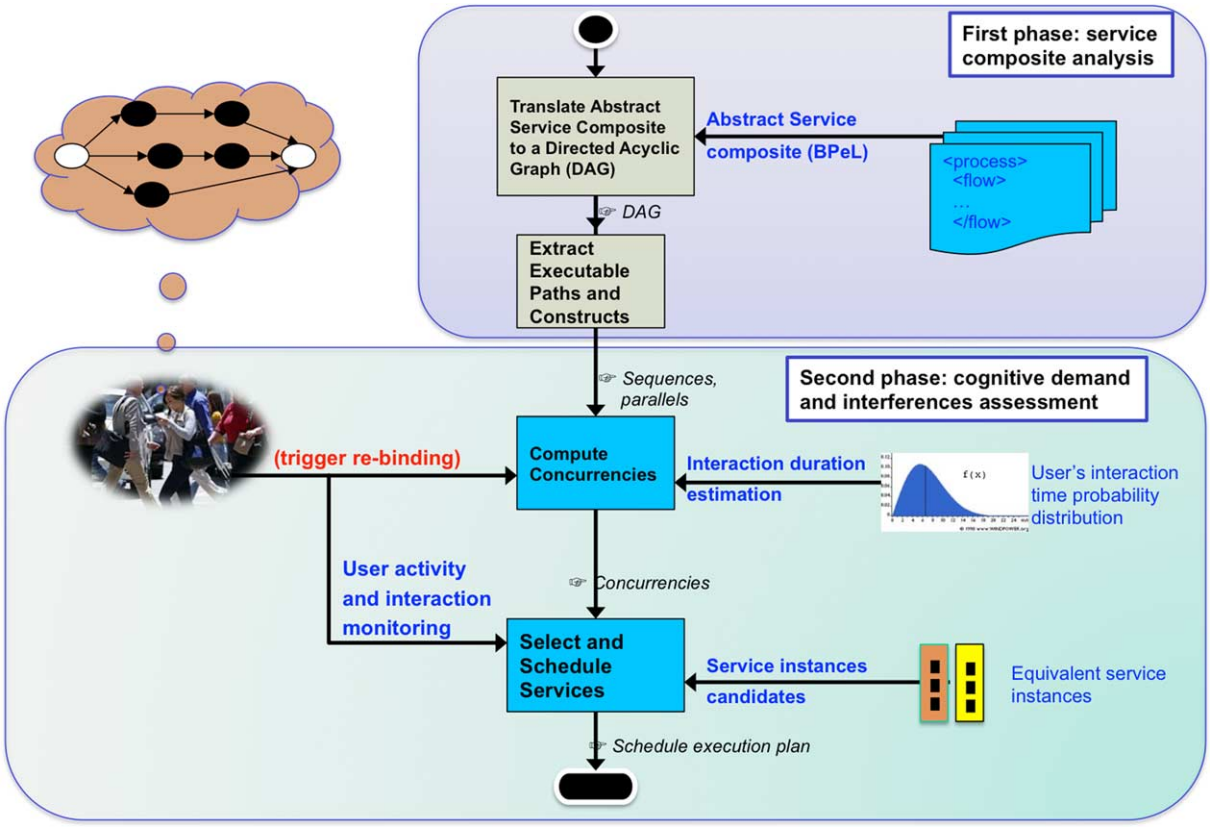

Fig. 4. Overview of the binding and scheduling mechanism.

with specific concurrent HCI services while engaged in a specific primary physical activity. These interaction times are stored by a monitoring module as historical data. The functions can be updated, and a fitted statistical model can be found at runtime by mining the historical data. To avoid cold-start data generation, a fitted model can be obtained either from collective intelligence about users' sessions with service compositions in AmI environments or from publicly available datasets, such as the Intel Computer Use Research report, which contains information on 263,612 user sessions with commonly used ubiquitous services [4].

Figure 4 shows that the mechanism dynamically (re)binds and schedules services by iteratively interleaving user interactions and service selections. All the steps run on a mobile client, except the *select and schedule services* step, which runs on the server side.

5.1. Service composition analysis

In the first phase, the BPEL file containing the service composition extracted from the service repository is translated into a directed acyclic graph (DAG) representation to make it easier to fulfill the coordination of abstract services. This DAG is used to extract executable paths in accordance with the control flow patterns that express the abstract service coordination. As shown in Fig. 5, executable paths represent all possible alternatives of the composition execution. Thus, sequential and/or parallel constructs are extracted from each executable path. Each of these constructs feeds the next phase to bind service instances to its abstract services.

The mechanism applies different types of binding and scheduling strategies depending on the structure of the composition, which is implemented in the second phase of the mechanism.

5.2. Cognitive demand and interferences assessment

5.2.1. Simple and sequential structure

Let us first refer to a chain of services that run in the background but that end in an HCI service as a *cohesive service sequence*. In contrast, let us also refer to a chain of cohesive, data-flow-dependent sequences as a *dependent cohesive sequence*.

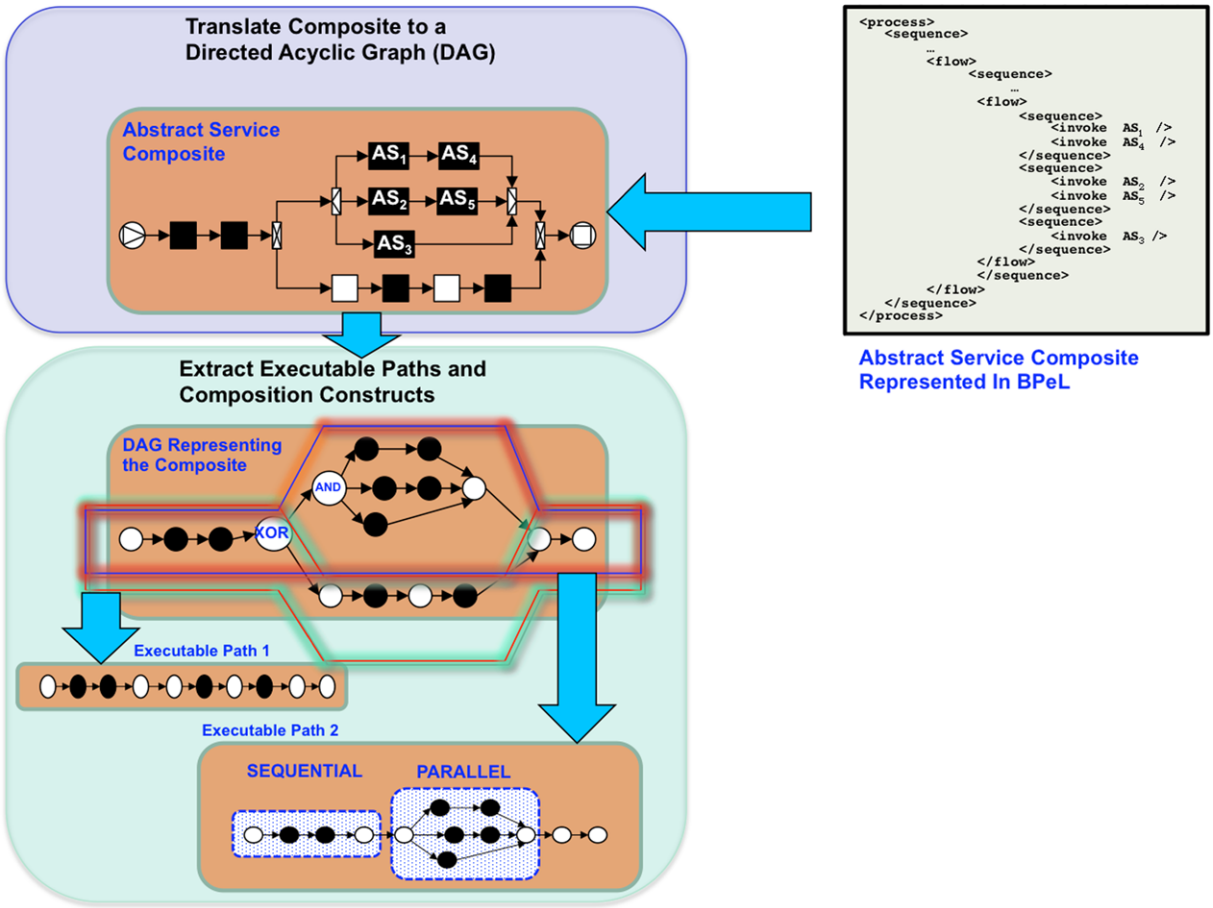

Fig. 5. Details of service composite analysis phase.

The simplest structure of a composition consists of a sequential behavior with one cohesive service sequence. In such a case, for each service instance candidate, the occurrence of input modality interference between the physical activities and the cohesive service sequence is initially checked. The second step assesses the processing stage and the conflicts in the processing code. Depending on the combination of values for these types of conflicts, the mechanism follows different conditional steps. For instance, if no interference is verified, the cohesive service sequence is a valid sequence. However, if partial interference exists, the mechanism computes the total amount of resources demanded by the cohesive service sequence in conjunction with all of the physical activities. Thus, the mechanism determines if this value exceeds the user's total processing capacity.

An extension of the simplest case is when there are data-flow dependencies among cohesive service sequences, which implies that they cannot be time shared. Otherwise, the dependencies, i.e., the order of precedence as stated in the composition, would be violated. However, this dependent sequential analysis is quite similar to the simple sequential behavior. Specifically, the analysis needs to be repeated for each of the cohesive service sequences.

5.2.2. *Deserialization of sequences without dependencies*

A different case consists of a series of cohesive service sequences without data-flow dependencies, that is, sequences that may be delivered independently. The goal in this case is to attempt a deserialization of the series of cohesive sequences for each service instance candidate. The rationale can be explained by considering the benefits of presenting concurrent functionalities that aggregately do not surpass the limit imposed by the user's total processing capacity. This is supported by the principle of progressive disclosure [47], which is widely used in human-computer interaction

and which states that "to help maintain the focus of a user's attention" [5], it is necessary to sequence "information and actions across several screens to reduce the potential of overwhelming the user" [5]. The logic behind this approach is "about ramping up the user from simple to more complex actions" [5]. Therefore, by applying this principle, there emerges a need to prioritize cohesive service sequences in terms of their cognitive demands. Once the serial cohesive service sequences have been ranked, a breadth-first algorithm identifies the service cohesive sequences that can be deserialized.

Details of the algorithms for sequential structures and deserialization are provided in the previous work [30].

5.2.3. Concurrent structure

In this paper, the previous work is extended by providing an optimization-based approach to solving the case of a concurrent structure. Moreover, using a parallel construct, the first step consists of computing its service concurrencies based on the probability density functions described above. This computation defines a set of concurrencies composed of the HCI abstract services that need to be bound at this iteration.

The problem of finding the best cognitive-resource-aware service composition is an optimization problem in terms of service concurrency. This is performed by extracting the cognitive dimension values from the CoS profile of each HCI task of each service instance to assess the total demand of mental resources and their interference.

Therefore, concurrencies are utilized as an input for a binary linear programming problem (BILP) [7] that attempts to select an optimal set of HCI service instances such that (1) the total mental resource demand by the HCI service instances (composed of the solution found by the solver) and the user's physical activity is minimized and (2) at each concurrency, no interference exists among the three dimensions of the mental resources demanded by the HCI tasks of the HCI service instances and the user's physical activity.

As explained in Section 3, according to the MRT, the total mental resource demand by multiple, time-shared activities can be calculated as the sum of the mental resources demanded by each activity. This assumption reflects the additive nature of the cognitive burden a user faces when performing time-shared activities [46]. Therefore, our model computes the total demand as the sum of the cognitive resources demanded by the physical activities and the HCI tasks. In turn, the resources demanded by the physical activities are the sum of the average resources demanded by each physical activity. The total demand of the HCI tasks functions in the same manner. All demands are computed using a three-level coding metric, which is adequate to account for the important variances of activity interference in accordance with [63]. Cognitive attributes have been extracted from [49].

By solving the BILP problem, the service selection mechanism identifies a set of background and HCI service instances that generate feasible, relaxed, or non-feasible solutions based on the availability of service instances in the search space. These service instances, their start times evaluated on the basis of their temporal latencies, and their estimated durations derived from the probability density functions define the scheduled execution plan to be orchestrated and are used to provide less cognitively taxing HCI service instances to the user.

5.3. Monitoring

A user's behavior is monitored by identifying any termination event of the user's interaction with the HCI service instances. Whenever a termination event is identified, the status of each service contained in the execution plan of this iteration is updated. Such a status may correspond to `terminated`, `scheduled but not executed`, or `running`. To ensure generality, more than one service instance can be catalogued using the `terminated` status. This is because even though the probability is low, it may be possible that more than one interaction with the service instances is terminated simultaneously.

A subsequent iteration of the mechanism is triggered with the set of `terminated` service instances, the set of `running` service instances, and the set of `scheduled but not executed` service instances. This new iteration consists of computing the new structure of concurrencies that emerge by considering the set of service instances whose status is either `scheduled but not executed` or `running` and the set of abstract services of the service composition that are the neighbors of the set of service instances with the `terminated` status.

5.4. Formulating the BILP problem

Let S be the set of abstract service types contained in the set of concurrencies of a specific iteration, whereby the available service instances must be bound and scheduled. An abstract service type t is denoted as S_t

and is composed of a set of service instances. In addition, let s_g^t be a service instance g of S_t. The following variables are defined to compute the cognitive resource demand:

- $U = (A_1, A_2, \ldots, A_n)$: A situation composed of activities A_i
- R: Totality of human cognitive system resources
- $R_u = (P_u, G_u, E_u) \subseteq R$: Cognitive resources required by the activities of the situation
- P_u: Perceptual cognitive resources required by the activities of the situation
- G_u: Cognitive resources required by the activities of the situation
- E_u: Responsive resources required by the activities of the situation
- $R_i \subseteq R_u$: Cognitive resources demanded by the activity A_i
- d_i^j: Demand of cognitive resource r_j by activity A_i
- $R_s = (P_s, G_s, E_s) \subseteq R$: Cognitive resources required by the service instance s
- $T_s = (t_1, t_2, \ldots, t_m)$: HCI tasks required by the service instance s
- $R_k \subseteq R_s$: Cognitive resources demanded by the HCI task t_k
- $d_j^{k,g,t}$: Demand of cognitive resource r_j by the HCI task t_k of the service g of the abstract service type t

On the other hand, the following variables are defined to compute the amount of cognitive interference:

- $input\_mod_i^a$: 1 if the activity A_i produces the input modality a to the user, 0 otherwise
- $input\_mod_{g,t}^a$: 1 if the service g of the abstract service type t produces the input modality a to the user, 0 otherwise
- $pro\_code_i^b$: 1 if the activity A_i has the processing code b, 0 otherwise
- $pro\_code_{g,t}^b$: 1 if the service g of the abstract service type t has the processing code b, 0 otherwise
- $pro\_stage_i^c$: 1 if the activity A_i has the processing stage c, 0 otherwise
- $pro\_stage_{g,t}^c$: 1 if the service g of the abstract service type t has the processing stage c, 0 otherwise

Finally, the decision variable for service g is defined as X_g^t such that $X_g^t = 1$ if this service instance is selected as the abstract service type S_t; otherwise, $X_g^t = 0$.

The objective function of the BILP problem consists of the minimization of the summation of the cognitive demands produced by the HCI tasks of each selected service instance X_g^t and each activity A_i in the situation U. Therefore, it is expressed as follows:

$$Minimize \overbrace{\left(\sum_{i:A_i \in U} \sum_{j:r_j \in R_i} d_j^i\right)}^{\text{Activity Demand}} + \overbrace{\left(\sum_{t:S_t \in S} \sum_{g:s_g^t \in S_t} \sum_{k:t_k \in T_S} \sum_{j:r_j \in R_k} d^{k,g,t} * X_g^t\right)}^{\text{HCI Tasks Demand}} \quad (1)$$

The service instances that are selected need to ensure that no cognitive interference is produced, which can be verified by adding the input modality, processing code, and processing stage set of constraints as follows:

$$\sum_{i:A_i \in U} \sum_{a \in IM} input\_mod_i^a + \sum_{a \in IM} \sum_{g:s_g^t \in S_t} \sum_{t:S_t \in S} input\_mod_{g,t}^a * X_g^t \leqslant 1, \quad (2)$$

$$\sum_{i:A_i \in U} \sum_{b \in PC} pro\_code_i^b + \sum_{b \in PC} \sum_{g:s_g^t \in S_t} \sum_{t:S_t \in S} pro\_code_{g,t}^b * X_g^t \leqslant 1, \quad (3)$$

$$\sum_{i:A_i \in U} \sum_{c \in PS} pro\_stage_i^c + \sum_{c \in PS} \sum_{g:s_g^t \in S_t} \sum_{t:S_t \in S} pro\_stage_{g,t}^c * X_g^t \leqslant 1. \quad (4)$$

To ensure that the summation of the cognitive demands from the HCI tasks of the selected HCI service instances does not surpass the limit of the human cognitive capacity represented by the threshold TD^*, the following cognitive demand global constraint is added:

$$\sum_{t:S_t \in S} \sum_{g:s_g^t \in S_t} \sum_{k:t_k \in T_S} \sum_{j:r_j \in R_k \subseteq R_s} d^{k,g,t} * X_g^t \leqslant TD^* - \sum_{i:A_i \in U} \sum_{j:r_j \in R_i \subseteq R_u} d_j^i. \quad (5)$$

It is worth noting that without a loss of generality, the activity demand term can be omitted in both the objective function and the cognitive demand global constraint.

To ensure the selection of one service instance per abstract service type S_t, the following allocation constraints is added to the formulation:

$$\sum_{g:s_g^t \in S_t} X_g^t = 1, \quad \forall t : S_t \in S. \tag{6}$$

Finally, the non-negativity constraints are added as follows:

$$X_g^t \geqslant 0, \quad \forall g : s_g^t \in S_t, \ \forall t : S_t \in S. \tag{7}$$

This BILP problem can be solved using any solver strategy, as explained in Section 6. This produces a set of selected, available service instances whose HCI tasks do not produce cognitive demands beyond the limit of the human cognitive capacity and do not produce cognitive interference among the cognitive resource demands. The mechanism includes a relaxation strategy, which allows second-best solutions that slightly surpass the maximum cognitive capacity. This strategy is applied when a solution cannot be found.

6. Technical evaluation

6.1. Implementation

The cognitive-resources-aware HCI abstract service binding and scheduling mechanism has been implemented as an independent module of the task-oriented service framework described in [31]. The mechanism has been applied to a set of various abstract service components. This has been performed by a simulation program written in Java version 1.6 and Android 4.1.2. The GraphML input capabilities of the Jung Java library version 2.0.1 has been utilized to implement the translation to a DAG [2]. The BILP problem is implemented using the open-source Lp-Solve optimization solver, Java library version 5.0 [13].

The open source Colt Java libraries version 1.2.0 [1] and the open-source SimJava Java libraries version 2.0 [28] have been utilized to implement the probability density functions of the user's interaction time with different HCI service types.

The observer pattern has been utilized to implement the iterative interaction between the *service engine* and the modules that realize the cognitive-resource-aware abstract service binding and scheduling algorithm. This "is a software design pattern in which an object, called the subject, maintains a list of its dependent, called observers, and notifies them automatically of any state changes, usually by calling one of their methods" [3]. Thus, the algorithm (denoted here as the observer) subscribes to any termination of a user's interaction with a running HCI service instance, as verified while monitoring the orchestration of the service instance execution plan by the service engine (denoted here as the subject).

The experiments are conducted on a MacPro server that runs the OSX Lion Server with a 3.2 GHz quad-core Intel processor and 8 GB of RAM. A Samsung Galaxy S3 smartphone is used as the client.

6.2. Data generation

Different sets of input data were generated to feed the simulation:

- A set of different HCI abstract service types to be used to generate the abstract service components.
- A probability density function per each HCI abstract service type, which is utilized to estimate the user's interaction time for each type.
- A set of abstract service components from the HCI abstract service types and general background abstract services.
- A set of background service instances and a set of HCI service instances for each HCI abstract service type and their respective cognitive attributes.
- A set of user activities and their respective cognitive attributes.

The first and second datasets are generated based on the *Intel Computer Use Research* report [4]. As mentioned in Section 5, this dataset contains information on 263,612 user sessions. A session is composed of the time spent using various service instances from different categories. These data were obtained by monitoring the behavior of 136 Android smartphone users through tracking software installed on their mobile devices.

The interaction time between a user and an HCI service instance is described by fitting a probability density distribution function to the empirical data for each abstract service type (see Table 2). To statistically evaluate the time usage patterns for each abstract service type (not the session composition), datasets for each service type are assumed to be independent of each other. The data treatment and analysis that were conducted are presented in the following:

1. For each abstract service type, only the sessions wherein the type is used are considered.
2. A basic summary of the data is obtained by plotting the kurtosis and skewness for each dataset.

Table 2
Probabilistic distributions obtained from the Intel dataset

Abstract Service Type	Distribution	Shape	Scale	Mean Log	SD Log
Communication	Weibull	0.8314	102.79	–	–
Browsing	Weibull	0.6858	192.91	–	–
Games	Weibull	0.6924	579.35	–	–
Home Screen	Lognormal	–	–	2.8485	1.35
Location Based	Weibull	0.7453	144.01	–	–
Media	Lognormal	–	–	4.4846	1.71
Other	Lognormal	–	–	2.8384	1.55

Based on this graph, three probability density distribution functions are chosen as the best-fitting candidates for the dataset, as proposed in [18].

3. The three candidate probability density distribution functions are tested in terms of their goodness-of-fit to the dataset based on the following criteria:
   (a) The Kolmogorov-Smirnov and Cramer-von Mises statistics are used to measure the similarity between the empirical distribution function of the sample and the cumulative function of the fitted functions.
   (b) The Akaike and Bayesian information criterion is used to measure the loss of information when the sample is represented by the candidate distribution functions.
4. With the above information, the best candidate is chosen, and its parameters are calculated using the maximum likelihood method (see Table 2). Figure 6 shows, as an example, the results of the fitting procedure applied to the session times of a specific gaming service. Note from the upper-left graph that the Weibull (0.69, 579.3) probability density function fits the empirical data shown in the superimposed histogram very well. This distribution is in agreement to the same extent when this statistical fitting is analyzed from the cumulative density function point of view, as shown in the bottom-left graph. The upper-right graph shows that the sample quantiles fit the theoretical quantiles reasonably well. Finally, the last graph shows that the theoretical and sample probabilities also fit perfectly.

One thousand random abstract service components were generated using the seven abstract HCI service types. Each abstract service component is formed by sequential, parallel, switch, or loop constructs.

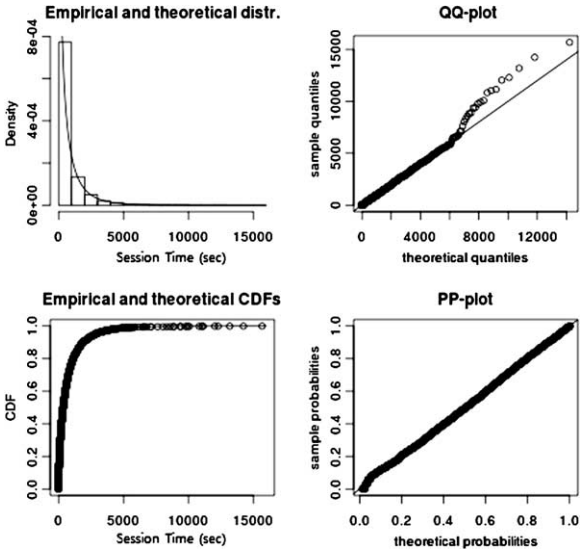

Fig. 6. Weibull (0.69, 579.3) probability distribution goodness-of-fit for data of a specific gaming service session time. Graphs from top to bottom and left to right: (i) Empirical data histogram and continuous Weibull distribution; (ii) cumulative density function (CDF) for the empirical data (circles) and the Weibull distribution (continuous line); (iii) quantile distribution for the sample data versus the quantile distribution for the Weibull fitted probability distribution, and (iv) observed probability distribution for the sample data versus the observed probability distribution for the Weibull fitted probability distribution.

For each of the HCI abstract service types, a set of 100 HCI service instances were generated, which were later decomposed into eleven HCI tasks with known cognitive attributes extracted from [49] following the encoding of Wickens mentioned in Section 5.2.3. The user behavior is represented as a user performing a set of physical activities in a given situation.

Two hundred test cases were defined as the combinations of two levels of the approach and 100 different sizes of available service instance candidates per each HCI abstract service type. The approach was divided in two levels as follows:

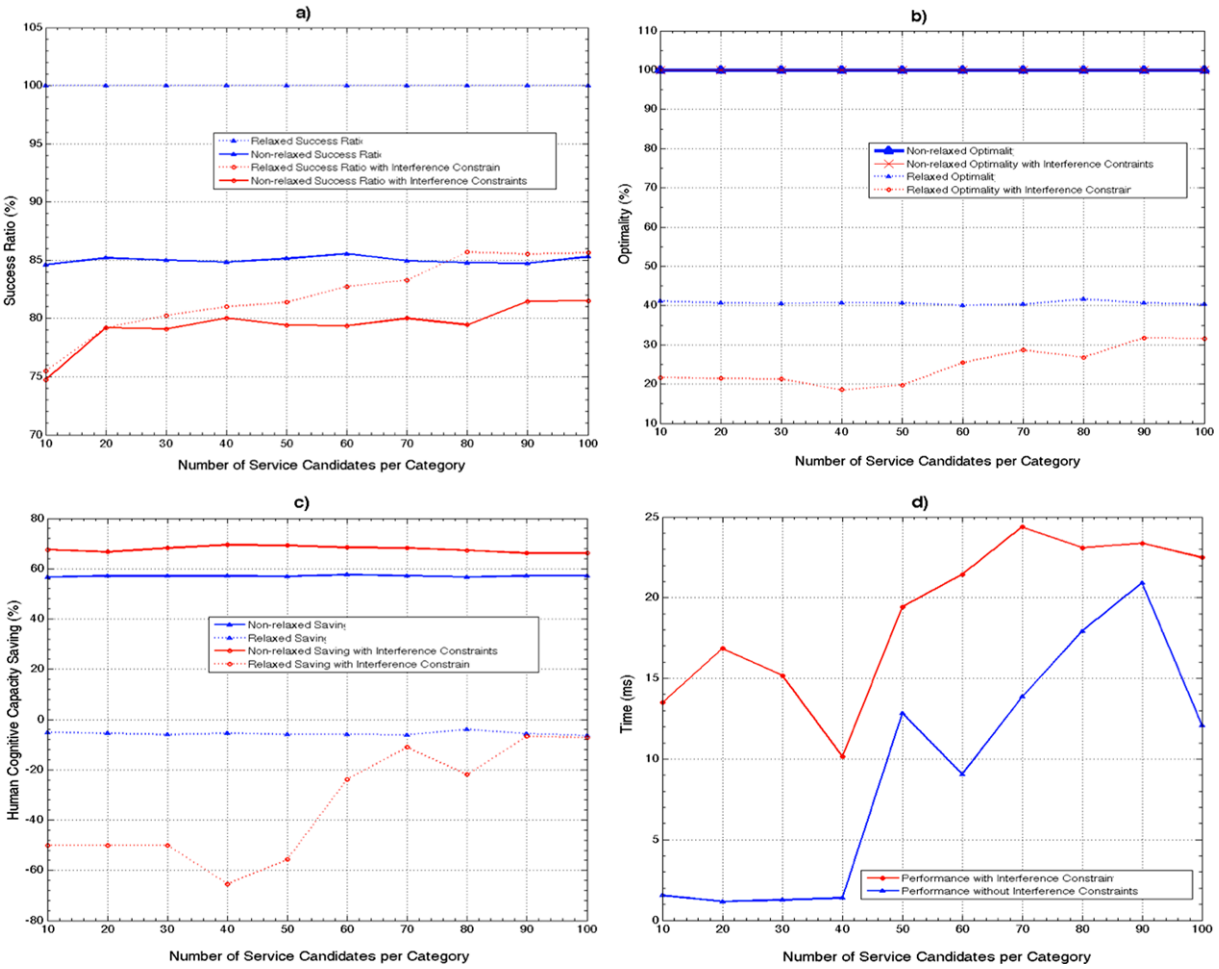

Fig. 7. Metric comparison before and after relaxation, with and without interference constraints: a) success ratio of concurrency bindings; b) optimality of concurrency bindings; c) human cognitive capacity saving; d) processing time with and without interference constraints.

- *Level 1*: Minimization of the cognitive demand aspect, constrained by the maximum cognitive demand.
- *Level 2*: Includes level 1 and is constrained by the cognitive interference aspect (input modality, processing code, and processing stages interference).

Table 3 shows the metrics considered for each aspect.

Table 3
Evaluation aspects and criteria

Aspect	Criteria
Cognitive Demand	– Success Ratio of Concurrency Bindings
	– Human Cognitive Capacity Saving
	– Optimality of Concurrency Bindings
Cognitive Interferences	– Input Modality Interferences
	– Processing Code Interferences
	– Processing Stage Interferences

6.3. Experimental results

Figure 7a shows the success ratio of the concurrency bindings. This is approximately the percentage of concurrency bindings that do not exceed the maximum threshold defined by human cognitive capacity. It is used the threshold defined in [63].

The success ratio for level 1 does not depend on the number of available HCI service instance candidates. For any number of candidates, the level 1 approach is able to find approximately 85% of the optimal solutions. However, this is at the risk of possible cognitive interference.

This figure also shows the success ratio that can be obtained if the relaxed concurrency bindings are also considered (100%). The relaxed concurrency bindings consist of those solutions provided by the BILP problem when the maximum human cognitive capacity threshold is relaxed. Specifically, these solutions are the second-best solutions that can be obtained with the overhead of over-demanding the human cognitive capacity of the user. This promising behavior of the level-1 approach means that solutions to the BILP problem exist. In this sense, the proposed relaxation strategy is demonstrated to be effective at this level.

Level 2 exhibits with a bigger risk of not finding any feasible solution and has a success ratio of between 75% and in some points is slightly superior to 80%.

In summary, even though this level completely avoids all the cognitive interference (input modality, processing code, and processing stage interference), there is overhead: a clear risk of providing infeasible, over-cognitively demanding concurrency bindings.

Figure 7b shows that the optimality of the service concurrency bindings that are relaxed to find feasible solutions decreases at level 2, while that of the relaxed concurrency bindings for the level 1 approach is steady at 40%.

The optimality using the level 2 approach barely achieves 30%. Therefore, at this level, the mechanism is not appropriate for providing acceptable service concurrency bindings because many second-best solutions are extremely cognitively taxing to the user. The optimality of the relaxed, second-best solutions is poor. However, these drawbacks are offset by the total elimination of cognitive interference.

Figure 7 c) shows the human cognitive capacity savings in percent. It is clear that the concurrency bindings obtained after the relaxation are over demanding. The human cognitive capacity savings of the concurrency bindings obtained using the level 1 approach before relaxation, i.e., using optimal solutions, nearly achieves 60% and behaves stably regardless of the availability of HCI service instance candidates. Theoretically, this means that the user only utilizes slightly more than 40% of his or her capacity in both the interaction with the HCI services and the performance of the physical activity. In contrast, the concurrency bindings obtained after the relaxation strategy are over demanding in all cases but never greater than a 10% extra cognitive demand on the user.

This is not a bad solution in the sense that, as demonstrated by Oulasvirta et al. [49], users are always able to define their own strategies to correctly address situations with low levels of excessive demand, such as by holding the phone and driving or biking and talking on the phone, without compromising the performance of their activities and interactions with the HCI services. Therefore, in terms of the criteria of human cognitive capacity savings, the level-1 approach is shown to be effective as well as in terms of the success ratio.

7. Discussion

This paper presents a novel, multidisciplinary approach to address the problem of CRD in the provision and consumption of unobtrusive service compositions in AmI environments. As shown in the recent literature, the technostress caused by this problem hampers users' engagement by draining mental resources, leading to a poor performance in both physical activities and HCI tasks. This interdependence between cognition and emotion justifies the examination of the cognitive context of a user as an integral part of the large umbrella posed by affective computing.

Physical activities and HCI tasks simultaneously demand multiple cognitive resources. This joint demand impinges on the limits of human cognitive capacity. In addition, the attributes of these resources may interfere with each other. Both the mental resource demand and interference produce a decrement in time-sharing efficiency when multiple activities are performed by a user. The relations among physical activities, HCI tasks and mental resources are represented in a cognitive-resources-aware model based on knowledge of the human information processing system and multiple resource theory. This model enables one to assess the cognitive resources demanded by HCI tasks and physical activities using the CoS profile. In addition, this model and its profiles enable one to determine the interference among multiple cognitive resource demands. The procedure used to identify cognitive resource demands and interference among their attributes for a specific activity or HCI task turns out to be a generalizable method because the CoS profile is generated in advance by conducting a task analysis. This is a practical approach, having considered that the state of the art of cognitive-resources-aware computing has yet to address the difficulty of assessing the cognitive processes of the human information processing system from the periphery. Current studies that make use of sensors – such as mobile eye trackers, electrocardiograms, and electroencephalogram headsets –

that capture cognitive loads bound to cognitive processes have been shown to only be meaningful in controlled settings [14,15,25]. They are still not generalizable to the diversity of physical activities and context changes in real-world scenarios.

In addition, this paper introduces a dynamic service binding and scheduling mechanism for the sequential and concurrent structures of a service composition. In this mechanism, the time window of the concurrent execution of abstract services is stochastically estimated in accordance with the probability density functions of interaction times. When the mechanism is applied to a real user, such functions should be trained from personal interaction data. This is something not shown, at least for now. However, this does not affect the validity of the measurements because a rigorous statistical fitting was conducted from experimental datasets. The same is valid for the user activities, which are extracted from a real-world time-use survey dataset. While randomizing service components guarantees variability of settings, it is also true that this procedure assumes that every setting is possible, which may not necessarily be true. To overcome this limitation, the set of service components was pruned by considering that humans can keep 7 ± 2 chunks of information in their working memory, as stated in [43]. To produce a conservative scenario, more than 5 branches in a parallel construct were not allowed.

One of the limitations of the technical evaluation is that the personal characteristics of a user cannot be captured, such as automaticity when using certain services and the influence of fatigue on the availability of cognitive resources. A user test to embrace such characteristics is required, which is something this research is currently performing and will be reported soon in the future work.

8. Conclusion

The major technical contributions of this work are as follow: (1) a cognitive-resource-aware activity and service description model has been described; (2) two theories from cognitive psychology have been transformed into a computational model; and (3) an effective and scalable algorithm for dynamically binding and scheduling service instances with abstract service compositions considering cognitive resources has been proposed.

As part of the framework for future research, there exists the opportunity to enrich this approach by adding new factors, such as emotion sensing [59], the automaticity that certain users may have when performing some HCI tasks, the influence that user fatigue may have on such performance, and the computation of mental workloads using multiple bio-sensors to measure physiological variables, that may affect the performance of time-shared activities to the binding and scheduling algorithm. In particular, this research is currently collecting data from a heart-rate monitor, a mobile head-mounted eye-tracker, an electrocardiogram, and a head-mounted electroencephalogram. The goal is to compare individual and hybrid classifiers based on their ability to assess mental workloads in a generalizable manner.

In addition, a more complete user study is currently being conducted by applying the NASA TLX mental load Questionnaire and the above physiological measurements [55]. Moreover, the mechanism is being integrated with a context manager to automatically trigger the optimization.

Acknowledgments

This research has been supported by CONICYT, FONDECYT INICIACIÓN under grant 11130252 and by Milenium Institute Complex Engineering Systems. This work was supported by the ICT R&D program of MSIP/IITP. [B0101-14-0334, Development of IoT-based Trustworthy and Smart Home Community Framework]. We thank all of our study participants.

References

[1] The Colt Project. [Online]. Available: http://acs.lbl.gov/software/colt/. Last accessed Nov. 10, 2014.

[2] The Java Universal Network/Graph Framework. [Online]. Available: http://jung.sourceforge.net/. Last accessed Nov. 10, 2014.

[3] The Observer Pattern. [Online]. Available: http://en.wikipedia.org/wiki/Observer_pattern. Last accessed Nov. 10, 2014.

[4] People and Practices Research, Intel Corporation. Intel Computer Use Research: Usage Tracking Data. [Online]. Available: http://www2.berkeley.intel-research.net/~tlratten/public_usage_data/pud.html. Last accessed Nov. 10, 2014.

[5] Progressive Disclosure. [Online]. Available: http://en.wikipedia.org/wiki/Progressive_disclosure. Last accessed Nov. 10, 2014.

[6] P. Adamczyk and B. Bailey, If not now, when?: the effects of interruption at different moments within task execution, in: *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 2004, pp. 27–278.

[7] M. Alrifai and T. Risse, Combining global optimization with local selection for efficient qos aware service composition, in: *Proc. of the 18th International Conference on World Wide Web*, ACM, 2009, pp. 881–890.

[8] M. Alrifai, T. Risse and W. Nejdl, A hybrid approach for efficient web service composition with end-to-end qos constraints, *ACM Transactions on the Web* **6**(2) (2012), 7.

[9] D. Ardagna and B. Pernici, Adaptive service composition in flexible processes, *IEEE Trans. Softw. Eng.* **33**(6) (2007), 369–384.

[10] B. Bailey, A framework for specifying and monitoring user tasks, *J. Comput. Human Behav.* **22**(4) (2006), 658–708.

[11] B. Bailey and S. Iqbal, Understanding changes in mental workload during execution of goal-directed tasks and its application for interruption management, *ACM Trans. Comput.-Hum. Interact.* **14**(4) (2008), 21.

[12] B. Bailey and J. Konstan, On the need for attention-aware systems: measuring effects of interruption on task performance, error rate, and affective state, *Computers in Human Behavior* **22**(4) (2006), 685–708.

[13] M. Berkelaar, K. Eikland and P. Notebaert, lp solve 5.5, open source (mixed integer) linear programming system, Software, GNU LGPL (Lesser General Public Licence). Available: http://lpsolve.sourceforge.net/5.5/. Last accessed Nov. 10, 2014.

[14] A. Bulling, D. Roggen and G. Troster, What's in the eyes for context-awareness? *Pervasive Computing, IEEE* **10**(2) (2011), 48–57.

[15] A. Bulling and T. Zander, Cognition-aware computing, *Pervasive Computing, IEEE* **13**(3) (2014), 80–83.

[16] R. Calinescu, L. Grunske, M. Kwiatkowska, R. Mirandola and G. Tamburrelli, Dynamic qos management and optimization in service-based systems, *IEEE Transactions on Software Engineering* **37**(3) (2011), 387–409.

[17] G. Canfora, M. Penta, R. Esposito and M.L. Villani, A framework for qos-aware binding and re-binding of composite web services, *Journal of Systems and Software* **8**(10) (2008), 1754–1769.

[18] A. Cullen and H. Frey, *The Probabilistic Techniques in Exposure Assessment*, 1st edn, Plenum Publishing Co., 1999.

[19] M. Czerwinski, E. Cutrell and E. Horvitz, Instant messaging and interruption: influence of task type on performance, in: *Proc. of OZCHI*, ACM, 2000, pp. 356–361.

[20] S. Draper, D. Norman and C. Lewis, Introduction, in: *User Centered System Design: New Perspectives on Human-Computer Interaction*, D. Norman and S. Draper, eds, Erlbaum, 1986, pp. 1–6.

[21] B. Dumas, D. Lalanne and S. Oviatt, Multimodal interfaces: a survey of principles, models and frameworks, in: *Proc. of Human Machine Interaction*, Springer, Berlin, Heidelberg, 2009, pp. 3–26.

[22] J. Fogarty, A. Ko, H. Aung, E. Golden, K. Tang and S. Hudson, Examining task engagement in sensor-based statistical models of human interruptibility, in: *Proc. of the SIGCHI Conference on Human Factors in Computing System*, ACM, 2005, pp. 331–340.

[23] M. Gil, P. Giner and V. Pelechano, Personalization for unobtrusive service interaction, *Personal and Ubiquitous Computing* **16**(5) (2012), 543–561.

[24] C. Groba and S. Clarke, Opportunistic composition of sequentially-connected services in mobile computing environments, in: *Proc. of the 2011 IEEE International Conference on Web Services*, IEEE, 2011, pp. 17–24.

[25] E. Haapalainen, S. Kim, J. Forlizzi and A. Dey, Psychophysiological measures for assessing cognitive load, in: *Proc. of the 12th ACM International Conference on Ubiquitous Computing*, ACM, 2010, pp. 301–310.

[26] J. Ho and S. Intille, Using context-aware computing to reduce the perceived burden of interruptions from mobile devices, in: *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 2005, pp. 909–918.

[27] E. Hollnagel, *Handbook of Cognitive Task Design*, Lawrence Erlbaum Associates, Mahwah, NJ, 2003.

[28] F. Howell and R. Mcnab, simjava: a discrete event simulation library for java, in: *Proc. of International Conference on Web-Based Modeling and Simulation*, 1998, pp. 51–56.

[29] R. Iqbal, J. Sturm, O. Kulyk, J. Wang and J. Terken, User-centred design and evaluation of ubiquitous services, in: *Proc. of the 23rd Annual International Conference on Design of Communication: Documenting and Designing for Pervasive Information*, ACM, 2005, pp. 138–145.

[30] A. Jimenez-Molina and I.-Y. Ko, Cognitive resource aware service provisioning, in: *Proc. of International Conference on Web Intelligence and Intelligent Agent Technology*, IEEE/WIC/ACM, 2011, pp. 438–444.

[31] A. Jimenez-Molina and I.-Y. Ko, Spontaneous task composition in urban computing environments based on social, spatial, and temporal aspects, *Engineering Applications of Artificial Intelligence* **24**(8) (2011), 1446–1460.

[32] W. Ju and L. Leifer, The design of implicit interactions: making interactive systems less obnoxious, *Design Issues* **24**(3) (2008), 72–84.

[33] E. Kim, S. Helal and D. Cook, Human activity recognition and pattern discovery, *IEEE Pervasive Computing* **9**(1) (2010), 48–53.

[34] J. Kong, W. Zhang, N. Yu and X. Xia, Design of human-centric adaptive multimodal interfaces, *International Journal of Human-Computer Studies* **69**(12) (2011), 854–869.

[35] J. Kreifeldt and M. McCarthy, Interruption as a test of the user-computer interface, in: *Proc. of the 17th Annual Conference on Manual Control*, JPL Publication, 1981, pp. 81–95.

[36] K. Latorella, Investigating interruptions: an example from the flightdeck, *Human Factors and Ergonomics Society Annual Meeting Proceedings* **40**(4) (1996), 249–253.

[37] J. Lee, J. Song, H. Kim, J. Choi and M. Yun, A user-centered approach for ubiquitous service evaluation: an evaluation metrics focused on human-system interaction capability, in: *Proc. of Computer-Human Interaction*, Springer, Berlin, Heidelberg, 2008, pp. 21–29.

[38] U. Lee, J. Lee, M. Ko, C. Lee, Y. Kim, S. Yang, K. Yatani, G. Gweon, K. Chung and J. Song, Hooked on smartphones: an exploratory study on smartphone overuse among college students, in: *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 2014, pp. 2327–2336.

[39] Y. Lee, C. Chang, Y. Lin and Z. Cheng, The dark side of smartphone usage: psychological traits, compulsive behavior and technostress, *Computers in Human Behavior* **31** (2014), 373–383.

[40] Y. Li, X. Zhang, Y. Yin and Y. Lu, Towards functional dynamic reconfiguration for service-based applications, in: *2011 IEEE World Congress on Services (SERVICES)*, IEEE, 2011, pp. 467–473.

[41] K.-J. Lin, J. Zhang, Y. Zhai and B. Xu, The design and implementation of service process reconfiguration with end-to-end qos constraints in SOA, *Serv. Oriented Comput. Appl.* **4**(3) (2010), 157–168.

[42] M. Miettinen and A. Oulasvirta, Predicting time-sharing in mobile interaction, *User Modeling and User-Adapted Interaction* **17**(5) (2007), 475–510.

[43] G. Miller, The magical number seven, plus or minus two: some limits on our capacity for processing information, *Psychological Review* **63** (1956), 81–97.

[44] Y. Miyata and D. Norman, Psychological issues in support of multiple activities, in: *User Centered System Design: New Perspectives on Human-Computer Interaction*, D. Norman and S. Draper, eds, Erlbaum, 1986, pp. 265–284.

[45] C. Monk, D. Boehm-Davis and J. Trafton, The attentional costs of interrupting task performance at various stages, in: *Proc. of the Human Factors and Ergonomics Society Annual Meeting*, SAGE Publications, 2002, pp. 1824–1828.

[46] D. Navon and D. Gopher, On the economy of the human-processing system, *Psychological Review* **86**(3) (1979), 214–255.

[47] J. Nielsen, *Progressive Disclosure*, Jakob Nielsen's Alertbox, 2006.

[48] A. Oulasvirta, T. Rattenbury, L. Ma and E. Raita, Habits make smartphone use more pervasive, *Personal and Ubiquitous Computing* **16**(1) (2012), 105–114.

[49] A. Oulasvirta, S. Tamminen, V. Roto and J. Kuorelahti, Interaction in 4-second bursts: the fragmented nature of attentional resources in mobile hci, in: *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 2005, pp. 919–928.

[50] S. Oviatt, Multi modal interactive maps: designing for human performance, *Human-Computer Interaction* **12**(1–2) (1997), 93–129.

[51] S. Oviatt, Human-centered design meets cognitive load theory: designing interfaces that help people think, in: *Proc. of the 14th Annual ACM International Conference on Multimedia*, ACM, 2006, pp. 871–880.

[52] L. Pessoa, On the relationship between emotion and cognition, *Nat. Rev. Neurosci.* **9**(2) (2008), 148–158.

[53] C. Roda, *Human Attention in Digital Environments*, Cambridge Univ. Pr., 2011.

[54] J. Rubinstein, D. Meyer and J. Evans, Executive control of cognitive processes in task switching, *Journal of Experimental Psychology: Human Perception and Performance* **27**(4) (2011), 763–797.

[55] S. Rubio, E. Diaz, J. Martin and J.M. Puente, Evaluation of subjective mental workload: a comparison of swat, nasa-tlx, and workload profile methods, *Applied Psychology* **53**(1) (2004), 61–86.

[56] M. Satyanarayanan, Pervasive computing: vision and challenge, *Personal Communications, IEEE* **8** (2001), 10–17.

[57] A. Schmidt, Implicit human computer interaction through context, *Personal and Ubiquitous Computing* **4**(2) (2000), 191–199.

[58] G. Tenenbaum, B. Hatfield, R. Eklund, W. Land, L. Calmeiro, S. Razon and T. Schack, A conceptual framework for studying emotions cognitions performance linkage under conditions that vary in perceived pressure, *Progress in Brain Research* **174** (2009), 159–178.

[59] E.L. van den Broek, M.H. Schut, J.H.D.M. Westerink and K. Tuinenbreijer, Unobtrusive sensing of emotions (use), *J. Ambient Intell. Smart Environ.* **1**(3) (2009), 287–299.

[60] I. Wassink, O. Kulyk, B. v. Dijk, G. v.d. Veer and P. v.d. Vet, Applying a user-centered approach to interactive visualisation design, in: *Trends in Interactive Visualization*, R. Liere, T. Adriaansen and E. Zudilova-Seinstra, eds, Advanced Information and Knowledge Processing, 2009, pp. 175–199.

[61] M. Weiser, The computer for the 21st century, *Sci. Amer.* (1991).

[62] M. Weske, *Business Process Management: Concepts, Languages, Architectures*, 1st edn, Springer Publishing Company, Incorporated, 2010.

[63] C. Wickens, Multiple resources and performance prediction, *Theoretical Issues in Ergonomics Science* **3**(2) (2002), 159–177.