Intelligent Environments 2017
C. Analide and P. Kim (Eds.)

doi:10.3233/978-1-61499-796-2-131

Designing an Intelligent Support Model of a Reading Companion Robot

Hayder M. A. Ghanimi[1], Azizi Ab Aziz[2] and Faudziah Ahmad[3]
Cognitive Artefacts Group, Human-Centred Computing Research Lab
School of Computing, Universiti Utara Malaysia
06010 Sintok, Kedah, Malaysia
hayder.alghanami@gmail.com,{aziziaziz, fudz}@uum.edu.my

Abstract. Human-functioning models that describe human cognitive and psychological states have been developed and used to serve as a core component in creating intelligent and responsive systems. Endowing such systems with these human-functioning models, it gives them the ability to reason and intelligently acting to attain its ultimate design object which is to assist and support people. The process to integrate these models with new systems remains as a nontrivial challenging task. This paper pinpoints the initial steps of integrating an earlier developed agent-based model of cognitive load and reading performance with a robotic lamp. In regard to this matter, several algorithms that have been developed are also discussed and simulations that were performed to prove the applicability of the proposed algorithms are presented.

Keywords. Dynamics Functioning Models, Robotic Lamp, Reasoning, Reading performance, Cognitive Load

1. Introduction

In the last several years, new technologies in building intelligent systems have improved tremendously and these systems have been integrated with human-functioning models to perform specific tasks. Human-functioning models are developed to understand complex cognitive and psychological behaviours of human. Therefore, by incorporating these models into software or machines, it will create intelligent applications that are capable of performing humanlike understanding and supporting behaviour [1], [2]. Examples of such models are the emotion contagion model in team members [3], aggression de-escalation model for Embodied Conversational Agents [4], and behaviour change model for hygiene and oral care [5]. Hence, these models have motivated a new attempt to come up with a model that can analyze cognitive load and relate to reading performance. This human-functioning model of cognitive load and reading performance has been developed as a building block for a proposed intelligent robotic lamp (reading companion robot). Equipped with this model, the robotic lamp will be able to understand readers' mental states and processes during demanding reading tasks as seen in [6], [7]. This paper aims to present the initial phase in materializing the integration process of the reader's model into a robotic lamp reasoning engine; where a number of algorithms were introduced.

The rest of this paper is structured as follows; Section 2 describes briefly the agent model of cognitive load and reading performance. Later, integration algorithm of the developed model with robotic lamp is introduced in Section 3. Section 4 discusses a scenario on how the algorithm works. Section 5 evaluates the proposed algorithms. Finally, Section 6 concludes the paper.

2. The Ambient Agent Model

To develop ambient intelligent applications (i.e., reading companion robot) it became necessary and highly desired to be based on modeling mental and physical aspects of humans. These models give clear insights and adequate knowledge to understand and analyze human behaviours. As a result, such applications will perform actions in a more informed, knowledgeable manner, and to show more humanlike behaviours in its interaction with humans [1]. Pertinent to readers' performances in demanding tasks, an ambient agent model of cognitive load and reading performance was developed to explain those processes. The details on our previous work can be seen in [7].

3. Components in Robotic Lamp

According to the pilot study that we conducted earlier, the table lamp was selected as a preferred object to be represented as a personalized medium of a reading companion robot when people read their books [8]. This reading companion robot has been designed by combining an Android smartphone that runs all computational units, table lamp stand as a base, Raspberry Pi microcontroller, and servo motors with four degrees of freedom into a reading-lamp based robotic system. The details descriptions of the robot and the synchronization process between software and hardware components altogether are beyond the scope of this paper. The pictorial idea illustrated in Figure 1 shows how the table lamp robot will be used in the reading environment.

Figure 1. The Pictorial Idea of Table Lamp Robot

4. Integration

One of the important concepts in modeling the dynamics of humans' physical and mental state is that these dynamics models can be used in real time to monitor and analyse related conditions to generate intervention actions [9]. Consequently, the software engine of this robot will be completely based on the ambient agent model. This model serves as an underlying analytical tool in monitoring and analysing readers' performance. Then, an appropriate support (as stored on support repository) will be suggested to readers. To attain this goal, an integration algorithm was developed to match appropriate supports with the analysed conditions. Figure 2 shows the central role of the integration algorithm. Further details on the ambient agent model of cognitive load and reading performance can be found in [6], [7].

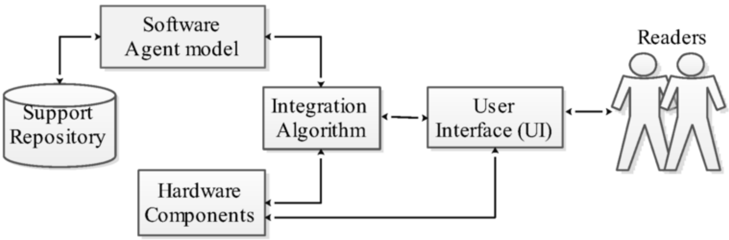

Figure 2. The Main Components in Reading Companion Robot

The generic flow chart of the integration processes is shown in Figure 3. It consists of six related components, namely; 1) input initialization, 2) environmental evaluation, 3) reader's monitoring phase, 4) evaluation mode, 5) support mode, and 6) external interruption by the reader to stop the monitoring mode (force stop mode). The processes will be described in the following sub-sections.

4.1. Inputs Initialization

At the beginning, the robot will initiate all of its observations to generate basic and derived beliefs related to reader's conditions. For example, when the robot observes the level of sound in the room, then it will compute its basic belief about the noise level.

$$observation(robot,\ sound(level)) \twoheadrightarrow basic\_belief(robot,\ noise(level))$$

In the end, all derived beliefs will be computed and later to be transferred to another component in the integration algorithm (as depicted in Figure 2). Next, we show some formal presentation and its nomenclature of important concepts used in our model.

Table 1. Formal Representations.

Descriptions	Formal
If robot *R* observes *x* condition then the robot will compute basic belief on *y*.	$o(R, x) \twoheadrightarrow b(R, y)$
If robot *R* believes on *y* then the robot will compute derived belief on *z*	$b(R, y) \twoheadrightarrow d(R, z)$
Robot *R* assesses the level of *x* whether it is greater or equal to x_threshold	$a(R, greater(x, xt))$ $a(R, equal(x, xt))$

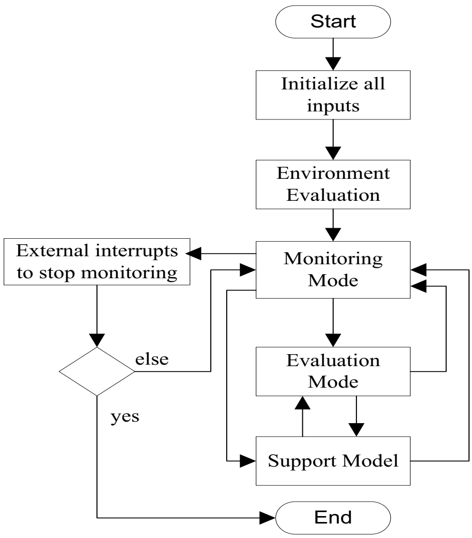

Figure 3. General steps in the Integration Algorithm

Algorithm 1 Environment Module
Input: $d(R, Pe)$
Output: ambient environment.
Start
  Initialization
    Pet, such that $0 \leq Pet \leq 1$
    Ab, such that $Ab \leftarrow$ false
  Evaluate physical environment
    Do
      if $(d(R, Pe >= Pet))$
        then $s(R, Cp_i)$
          if (Cp_i)
            then $v(R, Am)$
              $u(R, Pe)$
          else $s(R, Cp_{i+1})$
            if (Cp_{i+1})
              then $v(R, Am)$
                $u(R, Pe)$
            else $Ab \leftarrow$ true
      else $Ab \leftarrow$ true
    While (Ab)
  MonitoringMode( )
End

4.2. Environment Evaluation

Reading environment is a crucial factor that affects reading performance [10]. Therefore, our robot will evaluate the environment (i.e., physical environment based on the ambient agent model) before proceed to the next stage. Algorithm 1 shows the steps for the environment evaluation. It will analyse the derived belief of robot towards its physical environment, $d(R, Pe)$. If is the observed value is greater than the threshold Pet, then the robot will display a confirmation message to the reader to inform that the room is not ambience enough for reading process to take place, $s(R, Cpi)$, otherwise a monitoring mode will be activated. When the reader confirms the condition, the robot will advise the reader to perform some actions to reduce potential disturbance from the environment, $v(R, Am)$. Next, the derived belief of on the environment will be updated based on these new settings. The robot will display message again to confirm about the current condition of the environment. All of these steps will be iterated unless the reader makes a change to the ambience environment or he /she agrees with the current environment.

4.3. Monitoring Mode

The core part of the integration is a monitoring mode, where most of the analysis process occurs. The ambient agent (i.e., *the engine of the robot*) will be executed in parallel to compute all instantaneous beliefs and temporal assessments based on its inputs (*derived beliefs* and *initial values*) as in [7], and evaluate reading performance at the pre-determined time intervals. In details, this monitoring process includes:

I. Activate evaluation mode: when reading performance (*Rp*) is continuously decreasing ($dRp/dt < 0$) and it is lower than threshold *Rpt*.

II. Provide *praising dialogue*: for positive progress $p(R, Pg)$ if the reading performance is increasing and approaching the threshold $((Rpt - Rp) > mp)$ or maintaining it at $p(R, Pm)$ when changes in reading performance level is greater than threshold ($dRp/dt > 0$ and $Rp >= Rpt$).

III. Perform consistent checking $f(R, Cc)$: to ensure the model reflects real conditions. By doing this, the robot will display a confirmation screen $s(R, Cr)$ to evaluate a reader whether he/she experiences high cognitive load $e(r, H_{cl})$, high exhaustion $e(r, H_{ae})$, or low persistence $e(r, L_{pr})$ prior to provide an intervention part.

IV. Activate support exhaustion (*SupportAe()*): when the robot beliefs the readers have reached maximum hours of reading (based on predefined limit) (*Max_time*) and the reader should get a short break. Algorithm 2 presents the four main actions in monitoring mode.

Algorithm 2 Monitoring Module

```
Input: d(R, Tc), d(R, Tp), d(R, Tn), d(R, Pe),
       d(R, Pp), d(R, El), d(R, Pk), d(R, Rn), initial
       assessments, regulatory parameters, weight
       parameters, change rate.
Output: evaluated level of reading performance.
Start
  Initialization
    Rpt, such that 0 ≤ Rpt ≤1.
    c←2;   d←2;   k←2
  Repeat every time point t
    Compute all instantaneous beliefs ^ temporal
    assessments at timestep tp
    For every x, a(R, Rp) such that x ∈ tp
      if (tp/x = c-1)
         then if (dRp/dt ≤ 0 ^ Rp≤Rpt)
                then EvaluationMode( )
                   c ←c+1
             else c ←c+1
    For every y, a(R, Rp) such that y>x, y ∈tp
      if (tp/y = d-1)
         then if ((dRp/dt> 0) ^ (Rpt– Rp>mp))
                then p(R, Pg)
      if (dRp/dt >= 0 ^ Rp>= Rpt)
         then p(R, Pm)
      d ←d+1
    For every z, f(R, Cc) such that z > y, z ∈ tp
      if (tp/z = k-1)
        then s(R, Cr)
             if e(r, H_cl)
               then SupportCL( )
             else if e(r, L_pr)
                  then SupportPr( )
             else if e(r, H_ae)
                  then SupportAe( )
             k=k+1
    if time_spent ≥ Max_time
      then SupportAe( )
    tp←tp +1
    ts← ts + t
    Until reader stop monitoring
  End
```

4.4. Evaluation Mode

The robot will activate evaluation mode when reader's performance is low in order to detect the main factors that cause the undesirable level of reading performance. If the robot senses that the exhaustion level is continuously increasing ($a(R, dAe/dt \geq 0)$) between two intervals and exceeding the threshold *Aet*, then the support exhaustion model will be activated, otherwise the cognitive load and persistence conditions will be evaluated to provide the necessary help. Algorithm 3 shows the important stages for evaluation purposes.

Algorithm 3 Evaluation Module
Input: *a(R, Cl), a(R, Pr), a(R, Ae), a(R, Rp)*.
Output: determine the conditions that need support
Start
Initialization
Clt, such that $0 \leq Clt \leq 1$.
Aet, such that $0 \leq Aet \leq 1$.
Prt, such that $0 \leq Prt \leq 1$.
if $(a\ (R, dAe/dt \geq 0)\ \wedge\ a(R, Ae) \geq Aet))$
then *SupportAe ( )*
else if $(a(R, dPr/dt \leq 0)\ \wedge\ a(R, Pr \leq Prt))$
then *SupportPr ( )*
if $(a(R, dCl/dt \geq 0)\ \wedge\ a(R, Cl \geq Clt))$
then *SupportCL( )*
else if $(a(R, dCl/dt \geq 0)\ \wedge\ a(R, Cl \geq Clt))$
then *SupportCL( )*
End

4.5. Support Mode

Based on the robot's assessments and evaluations (monitoring and analysis), the support mode will be activated to provide the right support. For example, when the robot recognize that the reader has a high possibility of getting exhausted, then support exhaustion activity will be triggered (i.e., SupportAe()). This is important as exhaustion may deteriorate reading performance. The support mode has three main activities: Support exhaustion (*SupportAe()*), Support persistence (*SupportPr()*), and Support Cognitive Load (*SupportCL()*). The algorithms of these activities are presented as follows:

- *Support Exhaustion*

This type of support will be given in two conditions based on the robot's assessments as in monitoring and evaluation modes. First, when the robot observes and beliefs that the exhaustion level is continuously increasing and exceeding the threshold. Second, when the time spent for the reading task exceeds the predefined time (e.g., more than two or three hours) as mentioned in [13]. These procedures have been implemented in Algorithm 4 and specifically the algorithm decribes the important flows to support a potentially exhausted reader. Once this module is triggered, the robot will prompt a reader by a confirmation message $s(R, Cei)$ to rectify his /her exhaustion level. The short break advice $v(R, Sb)$ will be provided if the positive result has been detected. Otherwise, the second confirmation will be displayed to verify the actual exhaustion level. Next, when the support actions are activated, the robot has to reinitialize its monitoring mode based on default settings.

- *Support Persistence*

Low persistence is often positively correlated with the drop percentage of reading focus and continuation. However, motivational talks (short talk) will improve a reader's persistence. Our robot will activate a motivational talk module *p*(*R*, *Mt*) if a low persistence level has been observed. The algorithm to detect and provide motivational talk is presented in Algorithm 5. In the beginning, a confirmation message *s*(*R*, *Csi*) will be displayed to provide several selected motivational talks *p*(*R*, *Mt*). The robot will re-evaluate the condition upon receiving confirmation from readers. Then the derived belief on motivation will be updated when the reader receives a set of selected motivational talks as in *u(R*, *d*(*Mv*)).

- *Support Cognitive Load*

In this condition, readers with a cognitive overload condition will be supported by providing different approaches based on particular derived beliefs. These types of support can be viewed as:

- ❖recommends a similar task (*St*) when the derived belief on a reader's experience is lesser or equal to the threshold *d*(*R*, *El≤ Elt),*
- ❖provides specific knowledge (*Sk*) when the derived belief on prior knowledge *d*(*R*, *Pk≤ Elt)* is less than the threshold*,*
- ❖Advices on suitable materials *(Sm*) when the derived belief on task presentation is less than or equal to the threshold *d*(*R*, *Tn ≤Tnt*).

Algorithm 6 shows the important steps in supporting cognitive load. In this case, support actions should be prioritized related to the derived belief. For example, the lowest derived belief will be given a high priority in the support actions list. These processes also hold for the support cases in exhaustion and persistence. The robot will provide a particular support action from the list and also update its derived beliefs. In addition, the robot may suggest all support actions in parallel when the reader is experiencing a high cognitive load level.

Algorithm 4 *Exhaustion Module*
Input: *ts*, *tp*.
Output: To provide short break actions
Start
$s(R, Ce_i)$
if $s(R, Ce_i)$
then $v(R, Sb)$
$tp \leftarrow tp + 1$
$st \leftarrow tp + \mathrm{t}$
else $s(R, Ce_{i+1})$
if $s(R, Ce_{i+1})$
then $v(R, Sb)$
$tp \leftarrow tp + 1$
$ts \leftarrow ts + \mathrm{t}$
End

Algorithm 5 *Persistence Module*
Input: *d(R, Mv)*
Output: *p(R, Mt)*
Start
$s(R, Cs_i)$
if $s(R, Cs_i)$
then $p(R, Mt)$
$u(R, d(Mv))$
else $s(R, Cs_{i+1})$
if $s(R, Cs_{i+1})$
then $p(R, Mt)$
$u(R, d(Mv))$
End

```
Algorithm 6  Cognitive Load Module
Input: d(R, Pk), d(R, El), d(R, Tn)
Output: To support cognitive load.
Start
  Initialization
    i←1
  Compute Priority // assuming Cl=1
    if d(R, Pk ≤Pkt)
       then Sk← Cl *Pk
            List[i] ← Sk
            i←i+1
    if d(R, El ≤Elt)
       then St← Cl *El
            List[i] ← St
            i←i+1
    if d(R, Tn ≤Tnt)
       then Sm← Cl *Tn
            List[i] ← Sm
            i←i+1
  Bubble sort (List) // ascending
  if length.List≠ (∅)
     then s(R, Cd_i)
          if s(R, Cd_i)
             then for i← 1 to length_list
                  do g(R, List[i])
                     if List[i] = Sk
                        then u(R, d(Pk))
                     if List[i] = Sm
                        then u(R, d(Tn))
                     if List[i] = St
                        then u(R, d(El))
                     i←1
          else s(R, Cd_{i+1})
               if s(R, Cd_{i+1})
                  then for i← 1 to length_list
                       do g(R, List[i])
                          if List[i] = Sk
                             then u(R, d(Pk))
                          if List[i] = Sm
                             then u(R, d(Tn))
                          if List[i] = St
                             then u(R, d(El))
                          i←1
  else g(R, Sm^St^Sk)
       u(R, d(Tn^El^Pk))
End
```

5. Simulation Results

By incorporating an integrated ambient agent model of cognitive load and reading performance to a reading companion robot, this robot will be able to monitor and evaluates a reader's conditions based on certain observations. This section discusses how the ambient agent model works. Specifically, it details out the part on monitoring and assessment components using a scenario that relates to persistence of readers when reading difficult materials.

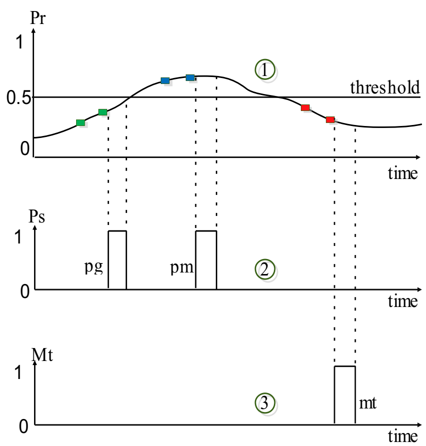

Figure 4. Persistence Support

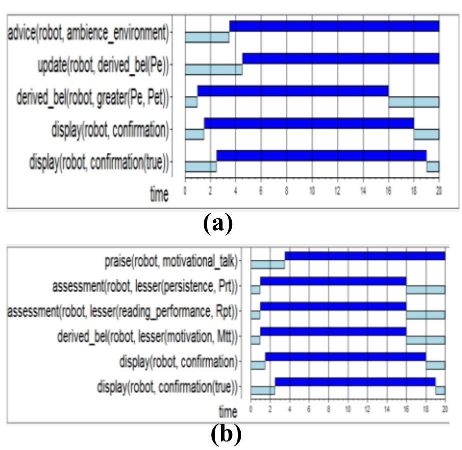

Figure 5. Simulation Results of (a) environment evaluation, (b) motivational talk

Figure 4 visualizes the description on monitoring persistence level (*Pr*) and provided support (i.e., praising (*Ps*) and motivational talk (*Mt*)). In this case, the persistence level at particular time will be acquired and determined the types of support. For example, when the persistence level is continuously increasing and approaching the threshold, a praising message for good progress will be displayed (*Pg*). Similarly, a praising cue to maintain the good progress will be displayed when the persistence level is greater or equal to the activation threshold (*Pm*). Likewise, a continuous and monotonic decreasing level in persistence triggers a motivational talk (*Mt*) mode. To visualize these processes, we have programmed and simulated a set of selected scenarios as depicted in the proposed integrated algorithms using LEADSTO platform. Figure 5 (a) and (b) show the simulation results. Based on the results, we can deduce that the simulation traces described above can satisfactorily explained the relations as summarized in Section 4.2, 4.3, 4.4, and 4.5.

6. Evaluation

This section describes the process to evaluate dynamic properties of the cases considered in the proposed algorithms. To do so, several possible cases were identified from the related literatures and an automated verification using Temporal Trace Language (TTL) was performed. TTL is also comparable to the *Holds*-predicate in the Situation Calculus. Based on that concept, dynamic properties can be formulated using a sorted First-Order Predicate Logic (FOPL) approach.

VP1: Provide social dialogues when the reading performance getting low [11].

∀γ:TRACE, t1, t2, t3:TIME, ∀M1, M2, D1, D2:REAL
[state (γ,t1) |= belief(robot, persistent_level(M1)) &
state (γ,t2) |= belief(robot,persistent_level(M2)) &
state (γ,t1) |= assessment(robot ,reading_peformance(D1) &
state (γ,t2) |= assessment(robot ,reading_peformance(D2)] &
t1 < t2 &M1 >M2 &D1 >D2] ⇒ ∃t3:TIME > t2:TIME [state(γ,t3) |=performed (robot, provide(social_dialogue)]

VP2: Suggestion to find an ambience place when noise level is high [10].

∀γ: TRACE, t1,t2, t3:TIME, N1, N2:REAL
[state(γ,t1) |= belief(robot, noise_level(N1)) &
state(γ,t2) |= belief(robot, noise_level(N2)) &
state(γ,t2) |= evaluation(robot, belief_ambience(no)) &
t1 < t2 &N1 > 0.6 & N1 ≤N2] ⇒ ∃t3:TIME > t1:TIME [state(γ,t3) |=performed (robot, advice(suggest_ambience_place))]

VP3: Advice for a short break session when the reader encounters exhaustion [12].

∀γ: TRACE, t1, t2, t3:TIME, E1, E2, d: REAL
[state (γ,t1) |= belief(robot, exhaustion_level(E1)) &
state (γ,t2) |= belief(robot, exhaustion_level(E2)) &
t1 < t2 + d &E1 ≥ 0.7 & E1 ≤E2] ⇒ ∃t3:TIME > t1:TIME [state(γ,t3) |=performed (robot, advice(short_break))]

7. Conclusion

The main goal of developing ambient agent models is to deploy technologies that are able to support people in their daily lives with human-like functionalities and understanding. Apart from this, the integration of an ambient agent model of cognitive load and reading performance into a reading companion robot is introduced in which integration algorithms were developed to serve as an intelligent mechanism for the robot's reasoning ability. For the next step, a thorough analysis of the proposed algorithms will be considered. This step is imperative in materializing the development of a robotic lamp for reading companion.

Acknowledgement

The project is partially funded by UUM Postgraduate Research Scholarship programmes.

References

[1] T. Bosse, M. Hoogendoorn, M. C. A. Klein, R. Van Lambalgen, P.-P. van Maanen, and J. Treur, "Incorporating human aspects in ambient intelligence and smart environments," Handb. Res. Ambient Intell. Smart Environ. Trends Perspec,, 2011, pp. 128-164, IGI Glob.

[2] J. Treur, "Making Smart Applications Smarter," in Network-Oriented Modeling, Springer, 2016, pp. 463–471.

[3] R. Duell, Z. A. Memon, J. Treur, and C. N. Van Der Wal, "Ambient Support for Group Emotion: an Agent-Based Model." In Agents Ambient Intell Achievements and Challenges in the Intersection of Agent Technology and Ambience Intelligence, , 2012, pp. 239–260.

[4] T. Bosse and S. Provoost, "Integrating conversation trees and cognitive models within an eca for aggression de-escalation training," in 18 International Conference on Principles and Practice of Multi-Agent Systems, 2015, pp. 650–659.

[5] O. Adegoke, A. A Aziz, and Y. Yusof, "designing a bdi agent reactant model of behavioural change intervention," J. Teknol., vol. 78, no. 2–2, pp. 83–93, 2016.

[6] H. M. A. Ghanimi, A. A. Aziz, and F. Ahmad, "An Agent-Based Modeling for a Reader ' s Cognitive Load and Performance," Adv. Sci. Lett. Appear., 2016.

[7] H. M. A. Ghanimi, A. A. Aziz, and F. Ahmad, "An Ambient Agent Model for a Reading Companion Robot BT - Computational Intelligence in Information Systems: Proceedings of the Computational Intelligence in Information Systems Conference (CIIS 2016)," S. Phon-Amnuaisuk, T.-W. Au, and S. Omar, Eds. Cham: Springer International Publishing, 2017, pp. 94–106.

[8] H. Mohammed, A. Ab Aziz, and R. Ahmad, "Exploring the need of an assistive robot to support reading process: A pilot study," International Symposium on Agents, Multi-Agent Systems and Robotics (ISAMSR), 2015. pp. 35–40, 2015.

[9] T. Bosse, M. Hoogendoorn, M. C. A. Klein, and J. Treur, "An ambient agent model for monitoring and analysing dynamics of complex human behaviour," J. Ambient Intell. Smart Environ., vol. 3, no. 4, pp. 283–303, 2011.

[10] H.-H. Choi, J. J. G. van Merriënboer, and F. Paas, "Effects of the physical environment on cognitive load and learning: towards a new model of cognitive load," Educ. Psychol. Rev., 2014, vol. 26, no. 2, pp. 225–244.

[11] J. Mumm and B. Mutlu, "Designing motivational agents: The role of praise, social comparison, and embodiment in computer feedback," *Comput. Human Behav.*, 2011, vol. 27, no. 5, pp. 1643–1650.

[12] A. Gulz, A. Silvervarg Flycht-Eriksson, and B. Sjödén, "Design for off-task interaction–rethinking pedagogy in technology enhanced learning," in *The 10th IEEE International Conference on Advanced Learning Technologies*, 2010, pp. 204–206.