Personalised time-dependent learning

R. Benlamri*
Department of Software Engineering
Lakehead University
955 Oliver Road, Thunder Bay
Ontario, P7B 5E1, Canada
Fax: +1 807 766 7243
E-mail: rbenlamr@lakeheadu.ca
*Corresponding author

Y. Atif
College of Information Technology
UAE University
P.O. Box 17555
Al-Ain, United Arab Emirates
Fax: +971 3 7672018
E-mail: Yacine.Atif@uaeu.ac.ae

J. Berri
Department of Computer Engineering
Khalifa University of Science, Technology and Research
P.O. Box 980
Sharjah, United Arab Emirates
Fax: +971 6 5611789
E-mail: berri@kustar.ac.ae

Abstract: Time-dependent instruction appears to shape next-generation learning systems, where the value of instruction is as important as the time it takes to learn. The ability to grasp the exact knowledge required to accomplish a specific task, in the allotted time, is a key factor for organisations to remain economically competitive in the new knowledge society era. Time-constrained learning is a new concept which requires a multilevel cognitive organisation of knowledge to suit various learning profiles. This paper proposes an ontology-based authoring tool capable of mapping concepts to learning resources to different granularity levels, hence customising learning-delivery to time-constrained learners. The proposed framework in this paper distils knowledge to meet timeliness using a judicious application of real-time systems principles. This interdisciplinary learning design advocates progressive levels of learning which trade learning granularity with allocated instruction time. The paper provides performance and experimental studies using an evaluation model and a validation approach through use cases. The results show interesting performance tradeoffs in analysing the cognitive perception of time-dependent learning.
Keywords: ontology; knowledge management; time-dependent learning; semantic web; adaptive learning; learning objects; learning web; personalised learning; learning technology.


Biographical notes: Dr. Rachid Benlamri is an Associate Professor in the Department of Software Engineering at Lakehead University in Canada. He earned a PhD in Computer Science from the University of Manchester, UK in 1990. His research interests are in the area of semantic web, mobile learning, semantic video analysis and distributed computing.

Dr. Yacine Atif is an Associate Professor at the College of IT at the UAE University. He earned a PhD in Computer Science from Hong Kong University of Science and Technology (HKUST) in 1996. His research interests include semantic web, e-learning, e-commerce and distributed computing.

Dr. Jawad Berri is an Assistant Professor of Computer Engineering at Khalifa University of Science, Technology and Research. He earned a PhD from Sorbonne University in France in 1996. His research interests include natural language processing, semantic web, agent technology and e-learning.

1 Introduction

E-learning standardisation efforts in conjunction with the advances made in the Semantic Web are expected to lead to synergistic effects and opportunities for sharing and reusing instructional resources globally in an open and intelligent learning environment. So far, most of the research in this field has focused on resource/service discovery and content adaptation problems based on learner context (Merceron et al., 2007; Mizoguchi et al., 2007; Davies et al., 2007; Denaux et al., 2005; Dolog and Nejdl, 2007; Benlamri et al., 2006). Very few researchers however have considered the time dimension in e-learning (Ozsoyoglu et al., 2004; Licandro et al., 2006; Papamichail and Papamichail, 2004; Berri et al., 2003; Maxim et al., 1999). Personalised learning however has several dimensions among which the time cost for learning that has not been explicitly addressed in the literature. This paper proposes a framework where the time dimension is integrated in the learning process to build next generation intelligent educational systems.

Learning effort incurs a time cost which may shift the outcome of learning beyond its utility levels or in some cases may even distract learners from an opportunity cost. For example, investing in foreign stock markets to cease a foreseeable business opportunity requires a time-limited knowledge acquisition about the foreign market and the business opportunity. Each delay in the learning process shrinks the benefits of the opportunity cost to the point where adverse consequences may result of committing for the original investment option as the dynamic situation of the market no longer reflects the previous learning domain. Hence a trade-off between the learning quality and the timely investment decision is prominent to optimal opportunity cost.
Another example of time-constrained learning is crisis management where community is educated to handle disaster situations. Governments or mayors may elaborate guidelines in the form of learning resources for handling unpredictable events that threatens national or communal interests. These resources provide means for managing the crisis such as assessing, understanding, and coping with possible serious situations, especially from the moment it first occurs to the point that recovery procedures start. Sudden Crises, such as fires, explosions, natural disasters, workplace violence, etc., provide limited time for action. The quality of the action depends on the quality of knowledge accumulated about the situation at hand. Hence, the volume of advocated knowledge will influence the outcome of the action to be undertaken. Volume of learning resources means also time to learn about the situation at hand, which may shift the (potential best) action beyond its expected utility.

The ability to grasp the exact knowledge required to solve a specific task, in a limited allotted time, is a key requirement for organisations to remain economically competitive in the new dynamically-evolving knowledge society. However, there is a lack of theoretical and empirical research in the area of time-dependent learning. New models and learning frameworks are essential to build learning environments that focus on progressive and multi-level order cognitive skills to allow time-flexible knowledge acquisition that can suit a variety of profiles. Many inefficiencies and waste exist in today’s Learning Management Systems (LMSs). Learning material often consists of large blocks of content and learners often need only small parts of that content (Bagley and Hunter, 1992; Brusilovsky, 1999; Castro et al., 2001; DeBra et al., 1999, Braun and Schmidt, 2006; Atif et al., 2003b). Furthermore, there is not much support in today’s LMSs for guiding learners in filtering out pertinent information in time-constrained learning environments. This contrasts with the latest trends in e-learning systems development suggesting to make e-learning pervasive throughout the enterprise’s ecosystem, delivering learning anytime, anywhere in a learner-driven context and on a learner’s schedule.

In this research, we propose an organisation of knowledge and a learning process to produce education that meets the immediate needs of time-starved learners with the ability to expand the grain of knowledge \textit{a priori} given a learning timeframe. This progressive approach to learning is achieved through the confluence of real-time systems principles, reactive planning concepts found in Artificial Intelligence, and the evolving Semantic Web based Learning standards. In this composite framework, learning material is automatically derived based on a time- and content-dependent function, where the instruction granularity is approximated to time cost. However, time-customisation of learning is difficult, especially if the learning material is not structured \textit{a priori} in multi-level cognitive degrees of depth, to enable progressive learning. This challenging goal is approached from an ontological mechanism to reason about the domain and produce Learning Webs (LWs) composed of atomic Learning Objects (LOs). These \textit{atoms} of learning are inferred by a unification engine to map grain-leveraged concepts to corresponding LOs resources. LW is a concatenation of LOs (or learning units) derived by an algorithm proposed in this paper and which produces a learning route to meet both instructional and timeliness specifications. The proposed ontology is used to facilitate both the authoring and construction of LWs so that learning content is ontology-conformant, and thus lends itself to be shared and reused.
While ontology-conformant learning content offers tremendous potential for content repurposing, the lack of standard-based reusable content (typically LOs with SCORM – Sharable Content Object Reference Model or LOM – Learning Object Metadata) that fits in a given learning design may be difficult to locate. Requirements for effective reuse of learning content in a learning design (Knight et al., 2005) based on an ontological framework include:

- control over granularity
- remove metadata that is irrelevant in the new context (example: objectives)
- each time an LO is reused, keep a record of how it was used to facilitate future recommendation (Atif et al., 2003b).

To address issues of granularity, relevant to time-based refinement of learning, we used an ontology framework for content repurposing to real-time contexts. In doing so, we store context-related metadata separately to capture granularity. The grain level of knowledge distillation can relate to the type or the semantic depth and breadth of the learning resource. Figure 1 illustrates an ontology which captures both types of granularity to provide alternative grains of learning. The proposed ontology construct integrates learning designs and LO content (Knight et al., 2005). The learning design is embedded in the hierarchy of the ontological links whereas the mapping of concepts to learning resources constrains the relevant LO content. Learning designs advocate pedagogical models and have already been shown to be flexible and reusable in various contexts (Knight et al., 2005), including real-time contexts. In this illustrative example, Information Retrieval concept can subsume varied degrees of breadth and depth sub-concepts, eventually leading to progressive coverage of learning content. In this paper, we argue that providing a judicious walkthrough algorithm, instruction quality can be mapped to a given learning design and selected learning content. We provide an ontology framework as knowledge base for time-constrained learners as well as database design of LOs which represent resource instances of related ontology concepts. Finally, we propose a matching algorithm which deliberates a LW based on a given time interval for instruction specified by the learner, to express the affordable slack time in the learning process.

The remaining sections of this paper are organised as follows. First we introduce the concept of design-to-time learning which maps knowledge quality to instruction time. We then describe an ontology-based LW authoring tool to segment instructional units based on standard LOs specification. Section 4 follows by revealing the system architecture of the proposed framework, after which a set of time-dependent learning-path construction algorithms are described. We then analyse empirically the performance of the suggested learning approaches and discuss a case study with respect to sample learners in using an implementation of the framework. Finally, we conclude the paper with a summary of results and future research related directions.
2 Design-to-time learning

As learning material scales up and longer instruction times become common, knowledge acquisition becomes inherently time-dependent. The quality of acquired knowledge is thus measured by the amount of essential or mandatory knowledge which might then be augmented with support knowledge to help in fulfilling learning outcomes if time permits. However, the utility of instruction may decrease with the allotted time should the learning-time cost outweighs the quality benefits. Thus, it is necessary to build a trade-off between the quality of instruction and the allotted learning time. By defining quality knowledge as a function which is proportional to the volume of supporting knowledge, we are able to maximise the amount of supporting knowledge while guaranteeing the acquisition of 'mandatory' (i.e., essential) knowledge. We refer to this process as design-to-time learning because the learning content is designed by the proposed system in this paper based on a time-to-knowledge mapping function. This mapping can be done statically (content-centred) prior to start a learning session by identifying all the LOs which cumulated durations meets a learner’s allotted instruction time. Time-to-knowledge mapping could also be performed dynamically (learner-centred) throughout a learning session by cumulating the durations of LOs as a learner progresses across a learning session. Time-dependent utility theory applied to learning technology in this paper is inspired from decision theory research (Horvitz and Rutledge, 1991; Zilberstein, 1993). In this theory, it is argued that the time to perform an action is as important as action outcomes. In our learning design case, it can be illustrated by learning practice to perform a time constrained action (for example attending a scheduled meeting with a client to demonstrate a new product or taking an exam at a specific instant in time). In this configuration, a learner may wish to plan the periods of time he/she would allocate to learn to meet the cut-off date of learning goal. Now given these time limits, we tailor learning volumes.
Existing approaches to learning delivery are either content-centred or learner-centred as mentioned above. The former is based on a content-driven instruction in which learners are exposed to pre-planned resources based on user-specified learning outcomes. This latter however, may result in multiple cognitive levels where learners regulate the depth/breadth of the knowledge construction process as well as the type of involved resources according to their needs and preferences. In a time-constrained environment neither approach is fully suitable to optimise the quality of the acquired knowledge in a preset time frame. The content-centred approaches limit the horizon of learning depth a priori regardless of the incurred time cost. Learner-centred approaches provide an in-depth learning experiences at an early stage of a learning session, and hence forcing time-constrained learners to progress across very shallow learning levels at the end of instruction. This is because most of the allotted learning time would be consumed at an early learning stage. There is no support to the learning process to guide instruction meeting a proportionate level of instruction to achieve a uniform depth ‘across the board’ for learners who have preset outcomes and fixed time slots. This paper introduces a trade-off between instruction time and the learning horizon while exploiting the benefits of both previous approaches to overcome their limitations in a time-constrained learning environment. The aim is to achieve a near-optimal learning quality in a situation where learners can afford a specific instruction time.

3 Ontology-based learning web construction

One of the most important pedagogical aspects in constructing time-enabled courseware is in adequately defining its scope and granularity levels, as well as the way it should be presented to the learner. The proposed framework combines learning outcomes with the time cost invested in achieving those outcomes. In a time constrained environment, traditional instructional designs are not suitable due to the rigid volume of resources proposed to the learner. The responsibility to filter out pertinent content and plan learning within the allotted time is not supported by existing instructional design systems. The main challenge however, is the lack of appropriate authoring tools which, guided by domain-related ontology could generate real-time conformant learning material. Different granularity levels are mapped to different time-costs to suit learners with different time profiles. We propose a multilayer model to relate ontological inferences to disparate web learning resources. The proposed model features an initial authoring process which structures learning content into two layers, namely instructional design layer (ontology layer) and authoring layer (LOs layer) as shown in Figure 2.

The ontology layer organises learning for a given domain. This layer includes the domain-related ontology which also controls both the semantic sequence and the grain level of the proposed content leading to a customised courseware referred to in this paper as Learning Web or LW. The authoring layer on the other hand, consists of two separate authoring activities: LOs and instructional design authoring. In a previous work where time cost for learning was not a factor, we combined these two authoring activities into a single activity whereby authors develop learning resources to build their own LWs (Atif et al., 2003b). In this work, the use of the ontology enables learning resources’ annotations using ontology concepts, and structuring the embedded instruction based on concepts’ sequence and granularity levels dictated by the ontology. Content producers
could develop their own LOs, or reuse already developed ontology-conformant LOs available on the web and accessible through exposed web services. Authors are further empowered with tools allowing them to search for, browse, retrieve, and store LOs, as well as LWs to create content. The LW construction process is a major module of the proposed system where ontology-related rules are used to guide authors in developing ontology-aware knowledge structure.

Figure 2  Conceptual model (see online version for colours)

3.1 Learning object structure

The ontology adjusts granularity of LO content to facilitate the authoring process. Granularity refers to the way authors map LOs to the concepts of the ontology. For instance, an LO may be very general, having a coarse grain size, if mapped to a higher-level concept in the ontology hierarchy or involving limited resources types. Another LO may represent inner grain concept that is mapped to a leaf node of the ontology hierarchy. The LOs used in this study follow the LOM (IEEE, 2002) standard specification. LOM introduces a base schema that abstracts data elements for LOs into metadata with nine categories. Below is a list of some LOM data elements used in our time-dependent learning system:

- **General.Title** – name given to this LO.
- **Technical.Location** – a string that is used to access this LO (might be a URL, for example).
• *Educational.TypicalLearningTime* – approximate or typical time it takes to work with or through this LO for the typical intended target audience.

• *Educational.SemanticDensity* – defines a measure of this resource’s usefulness as compared to its size and duration.

Our actual implementation of an LO makes use of a range of multimedia information as shown in Figure 3. To facilitate a third dimension of time-based learning design (given the depth and breadth traversal of the conceptual ontology mentioned earlier), we also adapted an abstraction model of Learning Object Content (Knight *et al.*, 2005). This approach actually depicts different types of content chunks in terms of their granularity. This ontology facilitates learning adaptation with respect to users’ learning devices since access to our time-dependent service from a computing device may involve different types of resources compared to say a mobile device. The later case represents the user experiment environment of the proposed time-based learning design as discussed later in the case study section of this paper. These chunks of LOs inherit all LO parent properties including the prominent time and relation attributes which are used in our time-dependent learning algorithms as shown in later sections of this paper.

**Figure 3** LO content structure ontology (see online version for colours)

The *Relation* feature of LOM standard specification is implemented through embedded hyperlinks within an LO which links an LO to another semantically related LO to form the logical sequence of a LW. The learning path is the actual route traversed by learners in their pursuit of knowledge construction. In subsequent sections of this paper, we propose an algorithm which analyses each LO prior to its playback to assist users in navigating a LW and hence advocate a learning path. This process is practically implemented through the display (or not) of navigational hyperlinks within a LO.
Although every LO has been authored with embedded hyperlinks referred to as correlations in this paper, only the appropriate correlations that satisfy the time constraints and do not lead to a cycle in the learning process are enabled. The proposed algorithm for activating correlations decides the actual volume of learning resources confined within an affordable temporal space.

3.2 Learning web construction

Ontology-based guidance for courseware authoring enables authors to state the minimum learning requirements to achieve the expected learning outcomes, and then further enhanced with support resources depending on the available time laxity. This is accomplished by applying a set of rules based on the relationships among the ontology concepts. Figure 4 illustrates a more elaborated framework of our sample ontology introduced earlier in Figure 1. The prerequisite relation establishes a sequence relation between a LO and its pre-requisite, while the necessary part-whole relation incites authors to include LOs associated to all the necessary parts (also called mandatory LOs in this paper) of a concept in order to repurpose content to a minimum instruction quality while expanding in it further based on instruction time laxity. Figure 4 presents a simple ontology for ‘information retrieval’ as an illustrative example in this paper (Baeza-Yates and Ribeiro-Neto, 1999).

In order to formulate the ontology logical inferences, we use logical predicates. The relationship \( \text{prerequisite}(c_i, c_j) \) involving two concepts \( c_i \) and \( c_j \), denotes that concept \( c_i \) is a required knowledge (concept) of \( c_j \) and needs to be covered first. Rule \( R1 \) in Table 1 establishes a sequence constraint represented by the relationship \( \text{before} \) between the LOs \( \text{LO}_k \) and \( \text{LO}_i \), representing respectively the concepts \( c_k \) and \( c_i \). For instance, in Figure 4, concept \( c_{11} \) is a prerequisite concept for \( c_{12} \) and \( c_{13} \). This is
represented by the actual fact $a1$ in Table 1. The predicate $LOforC(LO_i, c_i)$ denotes the association specifying that $LO_i$ is an LO representing concept $c_i$. When rule $R1$ is applied, fact $a2$ is inferred and added to the knowledge base.

Table 1  Ontology related rules and inferred facts

<table>
<thead>
<tr>
<th>Rules</th>
<th>Ontology related Facts</th>
<th>LW related Facts (inferred)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R1$: prerequisite($c_i$, $c_j$) $\land$ $LOforC(LO_i, c_i)$ $\land$ $LOforC(LO_k, c_k)$ $\rightarrow$ before($LO_k, LO_i$)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$R2$: necessarypartof($c_i$, $c_j$) $\land$ in($LO_i$, LW) $\land$ $LOforC(LO_i, c_i)$ $\land$ $LOforC(LO_k, c_k)$ $\rightarrow$ in($LO_k$, LW)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Facts</th>
<th>Ontology related Facts</th>
<th>LW related Facts (inferred)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a1)$</td>
<td>prerequisite($c_{11}$, $c_{12}$) $\land$ $LOforC(LO_{11}, c_{11})$ $\land$ $LOforC(LO_{12}, c_{12})$ $\land$ $LOforC(LO_{13}, c_{13})$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>before($LO_{11}$, $LO_{12}$) $\land$ $LOforC(LO_{11}, c_{11})$ $\land$ $LOforC(LO_{12}, c_{12})$ $\land$ $LOforC(LO_{13}, c_{13})$</td>
<td></td>
</tr>
</tbody>
</table>

| $b)$ | partof($c_{11}$, $c_5$) $\land$ partof($c_{12}$, $c_5$) $\land$ partof($c_{13}$, $c_5$) |

| $c1)$ | necessarypartof($c_{19}$, $c_8$) $\land$ $LOforC(LO_8, c_8)$ $\land$ $LOforC(LO_19, c_8)$ $\land$ $LOforC(LO_20, c_8)$ $\land$ $LOforC(LO_21, c_8)$ $\land$ $LOforC(LO_22, c_8)$ $\land$ $LOforC(LO_23, c_8)$ |
|       | in($LO_{19}$, LW) $\land$ $LOforC(LO_8, c_8)$ $\land$ $LOforC(LO_19, c_8)$ $\land$ $LOforC(LO_20, c_8)$ $\land$ $LOforC(LO_21, c_8)$ $\land$ $LOforC(LO_22, c_8)$ $\land$ $LOforC(LO_23, c_8)$ |

Part-of relationship $partof(c_k, c_i)$ represents the part-whole relationship where concept $c_k$ represents a knowledge component of $c_i$. For instance, in Figure 4, concepts $c_{11}, c_{12}$ and $c_{13}$ are part of concept $c_5$. These relationships are represented by the facts $b$ in Table 1. The relation $necessarypartof(c_k, c_i)$ represents the necessary part-whole relation where concept $c_i$ cannot be completely understood without covering concept $c_k$. This constraint is supported by rule $R2$ in Table 1. This rule guarantees that if $LO_i$, representing concept $c_i$ is in the LW (fact represented by the predicate in$(LO_i$, LW),
then all LOs \( LO_k \) representing concepts \( c_k \), which are necessary-parts of \( c_i \), must be in the LW. For instance, in Figure 4 concepts \( c_{19}, c_{20}, c_{21}, c_{22}, \) and \( c_{23} \) are necessary parts of concept \( c_8 \). This is represented by the facts \( c2 \) in Table 1. Since \( LO_8 \), which is the LO corresponding to concept \( c_8 \), is included into the LW as shown in Figure 5, then rule \( R2 \) ensures that \( LO_{19}, LO_{20}, LO_{21}, LO_{22}, \) and \( LO_{23} \), corresponding respectively to concepts \( c_{19}, c_{20}, c_{21}, c_{22}, \) and \( c_{23} \), are all in the LW; accordingly facts \( c2 \) are inferred and added to the knowledge base.

Although the system gives a default LW structure, resulting from the order generated by the depth-first traversal algorithm, this structure is not compulsory for the authors. Authors can rearrange LOs, and the rules provided by the system are automatically applied to ensure that these rearrangements do not violate the ontology relations. For instance, in Learning Web 1 (Figure 5), \( LO_{12} \) is covered before \( LO_{13} \), whereas \( LO_{13} \) is located before \( LO_{12} \) in Learning Web 2. Both schedules are possible since none of them violates the ontology rules. However, since concept \( c_{12} \) is a prerequisite for concepts \( c_{12} \) and \( c_{13} \), both \( LO_{12} \) and \( LO_{13} \) should be scheduled after \( LO_{11} \) as shown in the LWs of Figure 5. Also, authors have the flexibility to map ontology concepts to LOs at different granularity levels depending on the preset learning outcomes. Figure 5 shows different LWs that are generated from the same ontology. In Learning Web 2, the LO associated to Concept 3 is presented at a moderate grain size avoiding to consider its sub concepts in the second level of the hierarchy, whereas in Learning Web 1, it is described at the finest grain size considering all elements of the concept’s sub-tree.

The above-mentioned relationships and rules are just those needed to illustrate the example given in Figure 5. Other rules are used to deal with other types of correlations between concepts. In Figure 5, LOs are numbered according to the ontology concepts they represent. For each concept used in the LW, a pool of LOs describing such concept is first retrieved, and then an LO candidate is chosen to represent such concept amongst the retrieved pool using the most frequently used LO criterion.

The authored ontology-aware LWs are the inputs to the time-dependent knowledge construction algorithms to be described in the next section. The LWs as described in Figure 6 are made up of a collection of interlinked LOs of two types Mandatory Learning Objects (MLOs) suggested by the necessary-part of rules as described above; and
Secondary Learning Objects (SLOs) which are suggested by the rules. Thus, MLOs correspond to the set of ontology concepts representing the backbone knowledge of the domain. All MLOs must normally be visited by the learner to achieve the expected learning outcomes. SLOs correspond to ontology concepts representing complementary knowledge of the domain. SLOs may be added to the learning route or path traversed by the learner dynamically based on the learner interactions with the LOs such as requesting supporting material in the form of examples, self-assessment resources, case-studies, pre-requisite knowledge, etc. LWs are then annotated for use in a time-constrained environment. Each LO is labelled with its semantic density and learning duration time respectively as shown in Figure 6.

**Figure 6** Example LW with time and semantic density (see online version for colours)

The traversal of the LW is performed by exposing learners to the corresponding LOs. In a time-constrained environment however learners need to plan the traversal of the LW to guarantee the coverage of mandatory LOs and to adjust the quality of knowledge to available time. Therefore, we should guide the learner towards full coverage of MLOs as well as selected SLOs in a predefined time-frame. LOs which do not satisfy the time-cost requirement do not participate in the learning path construction process. Consequently, the algorithm decides which embedded hyperlinks to enable within an exposed LO to the learner based on the time left to an instruction session.

### 4 System architecture

The architecture of the proposed system as shown in Figure 7 consists of three main components namely, the Interaction Manager (IM), the LW authoring unit, and the time-dependent learning adapter. A detailed description of the LW authoring unit has been given in the previous section. Below we describe the remaining two system components.
4.1 The interaction manager

The IM module listens to learners’ initiated events and reacts accordingly. When the learner requests a new course, IM invokes remotely the selected course and creates an instance of the course LW. It also specifies the corresponding MLOs. It should also be noted that LOs used in a particular LW might be retrieved from different LO repositories geographically distributed over the web. Retrieved LO instances are stored in a local cache memory for fast access. When the learner initiates a new learning session, IM retrieves the current LO from the cache memory and proceeds to its execution by playing the embedded media. The learner can interrupt a learning session anytime which prompts the IM to store the learner current path and to update the list of visited LOs.

During the playback of an LO, the system listens and catches any event that can be triggered by the learner. When an LO invocation occurs, the scheduler is activated to augment the learning path with a new LO and the corresponding correlative LOs are concurrently cached for possible future invocation. Only the MLOs sequence is pre-determined by the learning system, whereas SLOs sequence are triggered by the user. A backward move from an SLO causes the system to remove its copy from the cache memory and will not be proposed to the learner in the rest of the current learning session. Correlative objects which have been visited once by the learner do not need to be presented all over again as it is assumed to be an acquired knowledge unit.

4.2 Time-dependent learning adapter

Time-dependent learning adapter represents the core of the system. It consists of three modules: the scheduler, the LO adapter and the learning path adapter. The scheduler is the most important component of the system. It guarantees the coverage of the required knowledge in the preset time. The scheduling algorithms are described in Section 5 of
this paper. Based on the visited LOs, the LO adapter adjusts the LO content in terms of the embedded linkage to correlative SLOs. The correlations, which are represented in an LO in the form of hyperlinks are shown or hidden according to whether the learner has already visited them or not, and whether visiting them will violate or not the deadline assigned to the learning session. The learning path adapter is the module responsible for constructing a learning path that is adjusted to the learner’s time constraints.

5 Time-dependent learning path construction algorithms

In this section, we provide a detailed description of the time-dependent learning-path algorithms that can be executed on the scheduler. To overcome the limitations of the content-centred approach and the learner-centred approach, we have introduced a hybrid approach as an alternative solution for higher quality time-dependent learning. This approach is illustrated in Figure 8 and employs both strategies in a tandem mode where the content-centred approach is adapted to deliver a larger LW in the sense that it represents a volume of knowledge which coverage requires slightly more time than the allotted learning time. The learner-centred approach is then deployed to ensure that the depth of the LW is monitored continuously.

In the LW, the MLOs represent the core knowledge and are always included in the learning path. It is therefore assumed, for the three learning approaches, that the time a learner can afford to spend on a learning session is always greater than or equal to the time required to visit all MLOs. The remaining time $T_r$ is used by the algorithms to involve further SLOs in identifying a self-adjusted learning path from the LW. A learner initiates the learning process by requesting a courseware which is represented by its associated LW. This request triggers the search for a learning path. The IM is based on a client/server framework where the server’s is to fetch the invoked LOs by the client application. A 4-tuple data-structure is maintained by the client application $(P, L_p, T_r, SD)$ which are described in Table 2. Also, each LO $O_i$ is structured as 3-tuple data structure $(O_i.t, O_i.SD, O_i.Correlation)$ which represents respectively the typical learning time associated with LO $O_i$, its semantic density, and the set of references (hyperlinks) to SLOs that can be directly invoked from $O_i$. Two types of events can occur during a learning session. A forward event which corresponds to either a move to the next MLO or an invocation of a correlative SLO, and a backward event which occurs when the learner moves backward in the learning path constructed so far.
Table 2  Algorithm’s data structure

<table>
<thead>
<tr>
<th>Label</th>
<th>Legend</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P$</td>
<td>A sequence of references to the LOs that can be invoked by the learner at the current time. Initially, $P$ contains all MLOs of the courseware.</td>
</tr>
<tr>
<td>$L_P$</td>
<td>The actual learning path, initially set to the empty set ($L_P = \emptyset$) but will be updated during a learning session by adding the invoked LOs.</td>
</tr>
<tr>
<td>$L_P^{(i)}$</td>
<td>The learning path issued from MLO $O_i$.</td>
</tr>
<tr>
<td>$T_r$</td>
<td>The remaining time to complete the learning session. This time is initially set to the allotted time $T_a$ from which is subtracted the time required to run all MLOs.</td>
</tr>
<tr>
<td>$SD$</td>
<td>The current semantic density level investigated by the learning path construction algorithm. $SD$ can be set to any value in the range $(0,..,4)$ which represents (very low, low, medium, high, very high) respectively. This is initially set to very high, meaning that we first start exploring supporting knowledge with high semantic density.</td>
</tr>
</tbody>
</table>

5.1 Content-centred learning path construction

The content-centred knowledge construction algorithm is designed in such a way that the produced learning paths warranty the provision of a variety of important concepts to support most MLOs. This is done by activating from the entire LW the hyperlinks referencing the supporting LOs that has the largest semantic density and fits in the allotted time. The remaining LOs are deactivated, as they do not meet the time constraint. The search space, as shown in the algorithm described in Figure 9, starts from the set of MLOs and then propagates progressively towards the farthest SLOs. The algorithm explores MLOs in an iterative deepening process across several consecutive semantic levels starting from the highest value. The algorithm terminates when no feasible LOs can be found within a specified semantic density. Hence the content-centred learning algorithm considers both the LOs timings as well as their semantic density levels when iteratively deepening the LW exploration. The produced learning path is then presented to the learner for navigation.

5.2 Learner-centred learning path construction

The learner-centred knowledge construction algorithm is solely dependent on the time the learner can afford to spend on a learning session. This learning algorithm is fully dynamic in the sense that no predefined LOs are initially planned, except the MLOs. The input to the system consists of the entire LW as designed by the courseware author, and the learner is given full control to control his/her learning within the allotted time. Supporting LOs are shown to the learner as long as there is enough time for them to be explored. The navigation algorithm is however designed in such away to ensure that all MLOs can be explored within the allotted time-frame. It also gives the learner the freedom to choose a number of additional SLOs based on the remaining time. In such a scheme, progressively, the complexity of the learning-path is cut down by pruning the alternative correlations which do not satisfy the time constraint.
Figure 9  Content-centred time-dependent learning path construction algorithm

1. $SD = MAX_{SD}$;
2. IF $SD < MIN_{SD}$ THEN STOP;
3. FOR EACH MLO $O_i$ DO $L_p = L_p \cup \text{Explore}(L_p, T_r, SD)$;
4. $SD = SD - 1$;
5. GOTO 2;

The Explore Routine is based on a Branch-And-Bound algorithm as described below:

\begin{verbatim}
OUT $L_p$ Explore (IN-OUT $L_p$, $T_r$, IN $SD$)
1. Queue = $O_i$  // $L_p$ Root
2. WHILE [Queue $\neq \emptyset$]
3.  Current_LO = Head (Queue);
4.  Delete (Current_LO, Queue);
5.  Successors_List = Current_LO.Correlation;
6.  FOR EACH $O_j$ IN Successors_List DO
7.      IF ($O_j \in L_p$)THEN
8.         INSERT ($O_j$, Queue)
9.      ELSE
10.         IF ($O_j \notin L_p$) AND FEASIBLE($O_j$, $T_r$, $SD$) THEN
11.             INSERT ($O_j$, $L_p$)
12.             INSERT ($O_j$, Queue)
13.             $T_r = T_r - O_j.t$;
14.         ENDIF
15.     ENDIF
16. ENDFOR
17. ENDWHILE
18. RETURN $L_p$
\end{verbatim}

Each LO must succeed the feasibility check in order to be added to the learning path. The feasibility check is performed as follows:

\begin{verbatim}
BOOLEAN FEASIBLE(IN $O_i$, IN $T_r$, IN $SD$)
1. IF ($O_i.t \leq T_r$ AND $O_i.SD \geq SD$) THEN RETURN TRUE
2. RETURN FALSE
\end{verbatim}

The algorithm, as described in Figure 10, starts by invoking the first LO in the learning sequence $P$. The invoked LO is analysed prior to its playback in order to select from the correlative LOs those that satisfy the time constraint and do not lead to a cycle (i.e., have not been visited so far). These correlations are reflected in our system through hyperlinks in the text script area of the LO. The hyperlinks of the considered LO are activated accordingly. A local copy of the invoked object is then created and inserted in the learning path, and its media content is visualised. The learner will then either invoke the next MLO or a feasible SLO, and the same process is reiterated for the newly invoked LO. In case of a backward move however, the revisited LO is further analysed against the
time constraint. The analysis might result in deactivating some correlative LOs due to the time consumed in visiting the previous SLOs. The process terminates when the final MLO is reached and its entire correlative learning sequence has been deactivated.

Figure 10  Learner-centred time-dependent learning path construction algorithm

\[
\begin{align*}
P &= \{O, \text{ where } O \text{ is MLO}\} \quad /\text{ P is initialized with the set of MLOs} \\
M &= P \quad /\text{ M is initialized with the set of MLOs} \\
T_a &= \text{Allotted Time} \\
T_e &= \sum_{O \in P} O.t \quad /\text{ T_e is the total time needed to play all MLOs} \\
T_r &= T_a - T_e \quad /\text{ T_r is the remaining time} \\
L_p &= \phi \quad /\text{ L_p represents the learning path which is initially empty} \\

\text{do} \\
&\quad \text{Listen to events} \\
&\quad \quad \text{switch (event)}{ \\
&\quad \quad \quad \text{case next:} \quad \text{Moving towards next MLO} \\
&\quad \quad \quad \quad \quad \text{invoke}(O) \quad /\text{ learning object } O \text{ is invoked by the learner} \\
&\quad \quad \quad \quad \quad P = P - \{O\} \quad /\text{ Remove } O \text{ from } P \\
&\quad \quad \quad \quad \quad \text{inset } O \text{ into } L_p \quad /\text{ Inset } O \text{ in the learning path} \\
&\quad \quad \quad \text{case forward:} \quad \text{Moving towards an SLO} \\
&\quad \quad \quad \quad \quad \text{invoke}(O) \quad /\text{ learning object } O \text{ is invoked by the learner} \\
&\quad \quad \quad \quad \quad T_r = T_r - O.t \quad /\text{ Time to play } O \text{ is subtracted from } T_r \\
&\quad \quad \quad \quad \quad \text{inset } O \text{ into } L_p \quad /\text{ Inset } O \text{ in the learning path} \\
&\quad \quad \quad \text{case backward:} \quad \text{Returning to the previously visited LO} \\
&\quad \quad \quad \quad \quad \text{invoke}(O) \\
&\quad \quad S = \{O \mid (O \in O_.Correlation) \text{ and } (O \notin L_p) \text{ and } ((T_r - O.t) \geq 0)\} \quad /\text{ S contains the correlative sequence of } O_\cdot \\
&\quad \quad \quad \quad \quad /\text{ which have not been visited so far and} \\
&\quad \quad \quad \quad \quad /\text{ which satisfy the time constraint} \\
&\quad \quad \text{Activate } S \quad /\text{ Show references associated to all LOs in } S \\
&\quad \quad \text{while } (S \neq \phi) \text{OR}(P \neq \phi) \text{OR}(O \notin M) \quad /\text{ While last MLO is not fully} \\
&\quad \quad \quad \quad \quad /\text{ exploited & there is at least an SLO that} \\
&\quad \quad \quad \quad \quad /\text{ satisfies the time constraint} \\
&\quad \text{end} \\
\end{align*}
\]

5.3 Hybrid learning path construction

The content-learner hybrid approach is proposed as an alternative for a better time-dependent learning, as was shown in Figure 8. The algorithm given in Figure 11 is an alternative implementation of the modified content-centred algorithm. The new implementation is deliberated given to show the extra step (instructions 16 to 23 in Figure 11) taken by the new content-centred algorithm to deepen the resulting learning path one more semantic level in the LW hierarchy. The learner-centred knowledge construction algorithm is then called with the resulting learning path as input search space.

This approach combines the content-centred and learner-centred dimensions of learning by offering higher quality information than that provided by the learner-centred approach. The proposed hybrid approach is also adaptive in the sense that it does not totally restrict the learner to some pre-selected knowledge that might not be of interest to him/her. The compromise solution adopted by the hybrid algorithm intends to balance the benefits of both approaches. This is done, as shown in the algorithm described in Figure 11, by first applying the content-centred algorithm to statically select the most important supporting knowledge that satisfies the time constraint. Another layer of new
supporting LOs is then added by decreasing the lowest semantic density resulting from
the statically generated LW one level down. The resulting LW, which obviously requires
more time than that can be afforded by the learner, is then used as the input search space
from which the learner-centred algorithm dynamically adjusts the depth of the LW
branches based on the learner choices and the allotted time for learning. Thus, the hybrid
approach not only optimises the quality of the presented information but also adapts to
the learner needs.

Figure 11  Content-learner time-dependent learning path construction algorithm

6 Performance evaluation and user experience

In this section, we evaluate the performance of the algorithms presented in Section 5
based on the following scenario. Consider the case of a business company which aims at
maintaining awareness among its retailers about competitive products in the market to
enable products comparison in the sales process to allow retailers giving sound advice to
customers. This company recognises the importance of maintaining a knowledgeable
staff and educating their sales force. Many such companies believe that by integrating ‘e-learning’ into their overall business strategy, they can turn consumer-education initiatives into revenue opportunities. However, employers are understandably reticent about asking employees to spend time away from their day-to-day responsibilities to learn a new skill or acquire knowledge about a new competitive product. Therefore, the business company adopts a time-dependent organisation of its educational assets to enable its sales channel to acquire knowledge in real-time situations. Consider a sales representative en route to a customer location. He realises that he is missing a key knowledge for his presentation about a new competing product in the market. With his Personal Digital Assistant (PDA), the representative accesses his corporate e-learning environment to gain the necessary insight, in real-time. A course fragment is available about the new product in the form of an LW. The graph representing the LW is shown in Figure 12 with its sequence of MLOs and SLOs associated with their individual expected service or execution time and semantic density level as defined by the LOs’ authors. Based on this scenario, we define a simulation methodology under which the proposed candidate algorithms are evaluated with respect to some performance metrics to be introduced in the following sections.

Figure 12 Experimental LW (see online version for colours)

### 6.1 Experiment design

For the purpose of the simulation study, two attributes, which are randomly generated, are assigned to each LO as shown in Figure 12; a duration of time corresponding to the LO execution time; and a semantic density level representing LOs’ relevance. The LW shown in Figure 12 is used as an input for the three algorithms presented in Section 5. The experiments are conducted under varying degrees of the allotted learning time $T_r$ and semantic density level $SD$. In our experiments $T_r$ ranges from 10 to 70 units of time whereas $SD$ ranges from 0 to 4 levels and is initially set to Level 4 (i.e., very high). The targeted performance metrics are Knowledge Quality and Knowledge Distribution which are defined as follows:
Knowledge quality $K_Q$ measures the quality of supporting knowledge (SLOs) which are consumed by the learner. Considering each consumed SLO execution time $t_i$ and semantic density $SD_i$, $K_Q$ is calculated as shown below. The knowledge quality $K_Q$ is an indication on the relevance of consumed supporting knowledge and the required learning time. The proposed algorithms strive to optimise the knowledge quality performance metric $K_Q$ by maximising the semantic density of the visited SLOs while minimising their overall durations. For instance, the knowledge quality provided to a learner who has consumed $n$ SLOs of semantic density $SD_k$ each, over a total execution time of let’s say $m$ units of time, is much better than that provided by a single SLO of the same semantic density $SD_k$ and an execution time of $m$ units.

$$K_Q = \sum_i \frac{SD_i}{t_i}$$

Knowledge distribution $K_D$ is an intuitive quality measure. It evaluates the proportion of the supporting knowledge distribution in the learning path with respect to the corresponding MLOs. This metric assesses the level of SLOs concentration which can be deep but rooted at few MLOs or shallow across several MLOs. The objective is to obtain the best performance trade-off which balances the depth and the breadth of supporting knowledge in the allotted time. In our experiments, $K_D$ is calculated as follows: $K_D = n_{MLO} \times n_{SLO}$, where $n_{MLO}$ is the number of MLOs from which supporting knowledge has been invoked and $n_{SLO}$ is the total cardinality of SLOs in the learning path.

Let’s assume that a hypothetical learner with unlimited time resources would visit the greyed LOs of the LW in Figure 12 forming his individual learning path. This is our reference learner based on which we measure in the next sections the performance of the candidate algorithms when the same learner traverses the same LW in a bounded time-resource environment.

### 6.2 Analysis of learning algorithms in a time-constrained environment

A simulation study was conducted to visualise the behaviour of the three algorithms in a strictly time-constrained environment. Therefore, the three algorithms were run using $T_r$ of 25 units of time, which represents approximately 1/3 of the time needed to consume all SLOs explored by our reference learner as shown Figure 12. In particular, we trace each of three algorithms considering the LW in Figure 12 as input and our reference learner as a hypothetical user which navigates the LW. We then analyse and compare the learning paths produced by each of the three algorithms.

#### 6.2.1 Content-centred learning algorithm

The content-centred time-dependent learning algorithm constructs a reduced LW based on the learner’s time constraints. The learner is then expected to navigate within the reduced LW boundaries. The algorithm traverses the list $P$ of MLOs and computes for each MLO its correlative SLO list. The SLOs are selected so that the total execution time of all SLOs is less than $T_r$ with optimum $SD$ level. When running the algorithm presented in Figure 9, the reduced LW shown in Figure 13 is obtained. Note that there is only one instance of a particular SLO in the LW.
Considering a learner who would visit all the correlative path of MLO₁, he would not be able to move beyond SLO₁ according to Figure 13. The LOs in the obtained learning path are a subset of the grayed LOs shown in Figure 12. The rest (uncoloured LOs) are SLOs which could contribute to the expected learning path. Based on the allotted time, the system was able to provide the learner with six SLOs supporting partly the first few MLOs encountered in the LW as shown in Figure 13.

6.2.2 Learner-centred learning algorithm

The learner-centred time-dependent learning algorithm allows a learner to navigate in the entire LW within the allotted time. Figure 14 displays the resulting learning path when running the algorithm shown in Figure 10 on the input LW shown in Figure 12 and the hypothetical learner represented by the corresponding coloured LOs. The obtained scenario in Figure 14 shows that the learner has been able to visit the whole correlative path of both MLO₁ and MLO₂ as well as SLO₁₄. However, since he has consumed all of the allotted time $T$, when visiting SLO₁₄, his learning path can only be completed with the rest of the remaining MLOs.
6.2.3 Hybrid learning algorithm

The hybrid algorithm is a combination of both algorithms. It makes use of an extension of the content-centred algorithm to build an enhanced LW and then the dynamic algorithm is used to monitor progressively the learner’s navigation within the previously generated LW. Figure 15 displays the resulting learning path when running the hybrid algorithm shown in Figure 11.

Figure 15 Learner path as generated by the hybrid algorithm (see online version for colours)

![Diagram showing learner path with SLOs not visited due to time constraint and SLOs excluded from the expected learning path]

Figure 15 shows two types of LOs:

1. visited LOs (greyed in Figure 15) belonging to the learning path of Figure 12
2. SLOs of the resulting content-centred LW which could not be visited due to time constraint (doubly circled in Figure 15).

At LO SLO2, $T_r$ expired. Therefore, all remaining correlations have been automatically hidden forcing the learner to visit only the remaining MLOs. Figure 15 shows clearly the superiority of the hybrid algorithm over the two other approaches. The Hybrid approach provides the learner with SLOs of better depth than those provided by the content-centred algorithm, and better breadth than those provided by the learner-centred algorithm. The hybrid algorithm thus strives to balance both depth and breadth of supporting knowledge.

6.3 Performance results of time-dependent learning algorithms

In a first experiment, we fix the allotted time for learning $T_r$ to 35 units, that is approximately half of the time needed to consume all supporting knowledge (SLOs) encountered by our reference learner (see Figure 12). We then quantitatively measure the performance of the three algorithms by comparing the resulting $K_Q$ and $K_D$ for each algorithm. In a time constrained environment, this experiment represents a typical learning scenario where the learner can only go through half of the supporting knowledge embedded in the course material. Table 3 reveals the obtained results for the normalised $K_Q$ and $K_D$. 
From the results shown in Table 3, we observe an improvement in learning quality for the hybrid algorithm over the content-centred and the learner-centred algorithms by a substantial margin of 8% and 18% respectively, whereas the knowledge distribution has significantly improved by a margin of 46% and 55% respectively. These improvements measure the efficiency of the hybrid algorithm in terms of the quality and the distribution of acquired knowledge under strictly bounded time constraints.

In a second experiment, we run the algorithms with varying degrees of $T_r$ ranging from 10 to 70 units. It should be pointed here that a $T_r$ of 70 units of time is sufficient to consume all SLOs that are encountered by our reference learner as shown in Figure 12. The aim of this experiment is to compare the substantial gain in terms of knowledge quality and knowledge distribution achieved by each of the three algorithms under different time-circumstances. Consequently, in each run, we compute $K_Q$ and $K_D$. The graphs shown in Figure 16 and Figure 17 reveal the obtained results for the normalised $K_Q$ and $K_D$ respectively. In both figures, the hybrid algorithm outperforms the content-centred and the learner-centred algorithms in terms of knowledge quality and knowledge distribution. It can also be noted that the gain achieved in terms of knowledge quality $K_Q$ is substantial for $T_r$ in the range [35–50] which corresponds to [50%–70%] of the time needed to consume all supporting knowledge (SLOs). The knowledge quality will then increase to 100% for all three algorithms when the learner is allocated a $T_r$ of 70 units, thus allowing him to consume all supporting knowledge. It can also be seen that the hybrid algorithm provides a better knowledge distribution compared to both content-centred and learner-centred algorithms at high values of $T_r$ as shown in Figure 17. This is because it provides the same breadth as the content-centred algorithm by planning $a$ priori the distribution of SLOs and then extends the depth to ensure an optimum knowledge distribution. It is also interesting to note that the learner-centred algorithm exhibits the worse knowledge distribution as it depends solely on the learner’s moves across the LW when deciding the distribution of SLOs.

The two previous experiments were run using the LW given in Figure 12, which is traversed by our hypothetical learner in a time-constrained environment. One may argue that the results given in the previous experiments might be biased by the spatial-temporal configuration of LOs involved in the LW of Figure 12, as well as the navigation choices made by the hypothetical learner. To demonstrate the efficiency of hybrid learning approach, we devised eight different learning paths traversed by different learners. This experiment shows the behaviour of the three algorithms under completely different navigational choices as shown in Figure 18. For these LWs, a typical learner would require a $T_r$ of 55 units to consume all SLOs. As stated above, the aim of this experiment is to compare the substantial gain in terms of knowledge quality and knowledge distribution achieved by each of the three learning algorithms under different time-circumstances and different spatial-temporal LW configurations. Consequently, in each run, we compute $K_Q$ and $K_D$. The graphs shown in Figures 19 and 20 reveal the obtained results for the normalised average $K_Q$ and $K_D$ respectively. In both
figures, the hybrid algorithm outperforms the content-centred and the learner-centred algorithms in terms of knowledge quality and knowledge distribution, hence confirming the previous results.

Figure 16  Knowledge quality versus $T_r$, (see online version for colours)

Figure 17  Knowledge distribution versus $T_r$, (see online version for colours)

6.4 User experience

A case study was conducted among school students in mobile learning contexts, where time-dependent learning situations may inherently occur. In this environment, mobility is frequently suspended by fixed periods of stagnant states in particular locations. During these opportune occasions, often accompanied with inactivity, learners were invited to estimate the idleness period and invoke our learning service online. This could be a positive attitude to instil today’s young and mobile generation to usefully exploit their portable devices beyond the traditional (and not so useful) SMS chat! Typical stagnant overhead times could include a period of time spent waiting for or aboard a public transportation, or visiting waiting rooms (for medical or other reasons), etc. In this user experiment, students were equipped with mobile devices which were configured as clients of our time-constrained learning service. Figure 21 shows typical screen shots of a candidate student’s mobile device for the purpose of this case study.
Figure 18  LWs using different spatial-temporal configurations (see online version for colours)

Figure 19  Average knowledge quality versus $T_r$ (see online version for colours)
Initially, the mobile learner is prompted to select a domain of study and hence the related ontology. In this case, the learning domain relates to C++ Programming Fundamentals. Then the learner provides an estimated time, he is prepared to spend for the advocated instruction, based on his contextual situation. These two pieces of information triggers the hybrid algorithm shown in Figure 11, which has been established experimentally to outperform the other proposed algorithms in providing a LW that has best learning quality threshold. Depending on the size of the ontology, a learner may further select a particular subset of the ontology he would like to focus on. This additional information further constrains the scope of the learning domain and reduces the focus of instruction (Atif et al., 2006). In this Pull-technology scenario of mobile learning, learners are immediately driven to a volume of content to suit their cognitive and time constraints needs as shown in Figure 22. The learner is aware beforehand about the expected instruction time required to cover the lesson, unlike traditional mobile learning methods. Hence, learners can better plan their self-instruction performance.
In the initial survey collated from users of the above mobile learning application, which uses our time-dependent learning algorithm, it appears a positive apprehension of the cognitive perception of time proposed in this paper. The association of time to instruction has rarely been explicit in instructional technology solutions, prompting learners to make their own time-planning solutions. Yet, learners are neither able to filter-out prominent concepts nor are they aware about the overall scope of domain-related resources. These shortcomings limit the ability of learners to delimit the scope of their learning to fit an allotted time. The respondents of the user survey did highlight the positive impact of lifting away this dilemma and relying on the expert-adjusted learning assistance through the provided time-aware instruction software. This solution delivers a quality of instruction, which has been shown experimentally to be optimal, in the affordable instruction time.

7 Conclusion

In this paper, we argued that e-learning systems are required to exhibit a predictable behaviour in which the semantic value of learning is linked to the time it takes to reach the expected learning outcomes. We demonstrated the feasibility to make knowledge quality of a learning process constrained to the time it takes to acquire that knowledge. This is achieved by a novel approach to learning which integrates the allotted learning time and knowledge-quality as performance criteria of the learning process. The proposed system is based on an ontology-based LW authoring tool and real-time software development principles. The main contribution of the paper is a technique which optimises a design-to-time learning function that maps the volume of knowledge resources to time costs. To construct this function, we introduced a model of knowledge organisation based on which learning durations can be measured to accurately represent various learning scenarios. Three learning-process construction approaches were investigated. In the first approach, a learning-path is statically determined before exposing any knowledge to the learner, while the second approach constructs
dynamically a learning-path during the learning session. Finally, the hybrid approach combines the two previous strategies by delivering statically a learning-path while monitoring dynamically the learning session to further adjust the learning scenario in order to best meet the expected learning outcomes in the allotted user-defined time. The three approaches were compared based on semantic and temporal performance criteria. The experimental results have shown interesting performance trade-offs in assessing the time-constrained algorithms presented in this paper. Future research work consists in formalising our design-to-time learning approach to investigate further context related constraints in Mobile Learning. The challenge with the latter paradigm is to manage the learner’s context and to integrate, in addition to time, a set of system-centric constraints inherent to mobile learning.

References


