May 2006, Vol.21, No.3, pp.418–429 J. Comput. Sci. & Technol.

Checking Content Consistency of Integrated Web Documents

Franz Weitl and Burkhard Freitag

Department of Mathematics and Computer Science, University of Passau, D-94030 Passau, Germany

E-mail: {Franz.Weitl, Burkhard.Freitag}@uni-passau.de

Received January 25, 2005; revised March 20, 2006.

Abstract A conceptual framework for the specification and verification of constraints on the content and narrative structure of documents is proposed. As a specification formalism, $\mathsf{CTL}_{\mathcal{DL}}$ is defined, which is an extension of the temporal logic CTL by description logic concepts. In contrast to existing solutions this approach allows for the integration of ontologies to achieve interoperability and abstraction from implementation aspects of documents. This makes $\mathsf{CTL}_{\mathcal{DL}}$ specifically suitable for the integration of heterogeneous and distributed information resources in the semantic web.

Keywords document verification, content consistency, model checking, temporal description logics, CTL, $\mathcal{DL}$

1 Introduction

A large amount of information in the web is available in the form of web documents.

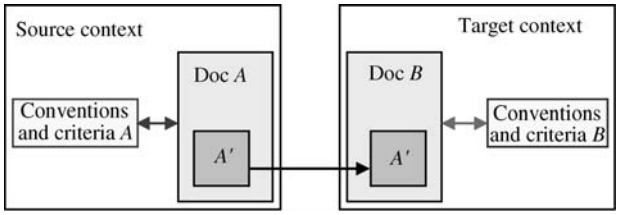

Fig.1. A document fragment is integrated into a new document: does the integrated document (still) conform to the set of conventions and criteria of the new context?

When integrating external resources into a document, its overall *semantical consistency* needs to be maintained (Fig.1) where *semantical consistency* is understood as the conformance of a document to *conventions and criteria* which apply within a certain *authoring context*.

In a scientific setting, for instance, the document should conform to a specific global structure (introduction — problem description — solution — discussion — conclusion) or to a certain argumentation style (a new concept should be well defined before it is discussed).

Especially when integrating external fragments from heterogeneous and autonomous sources the maintenance of the document's overall semantical consistency is a challenge. Complex documents such as web-based trainings or technical manuals are usually created in a team (collaborative development): will the integrated result be globally consistent in terms of structure and content? External resources may be changed by the owner over time (change management): do the changes make sense in all documents which the resource is integrated into?[1,2] Documents may be assembled automatically according to a specific information request[3] or adapted to the user's preferences[4] (cf. adaptable and adaptive hypermedia): is the document semantically consistent within the entire adaptation space? In these scenarios the automated checking of certain semantical consistency criteria is highly desirable[1].

In this paper we present an approach to representing relevant types of conventions and criteria in a *formal specification* and verifying the specification against a *model* representing relevant aspects of the document's content and structure. In contrast to existing methods of hypermedia verification our approach allows for the adoption of terminological background knowledge represented in formal ontologies. This increases the expressivity and interoperability of the specification formalism. Moreover, global properties of integrated documents can be checked even if based on globally inconsistent knowledge representations. This makes our approach particularly suitable for integrating heterogeneous and distributed web resources.

The rest of the paper is structured as follows: first we will summarize existing approaches in hypermedia verification and argue why more has to be done to be able to check the semantical consistency of a document. Subsequently we will give a short overview of our approach followed by a formal definition of the involved data structures and algorithms. After a brief discussion of our framework we conclude with a summary and outlook.

2 Related Work

A hypertext verification method based on temporal logic has been developed by Stotts, Furuta *et al.*[5]. A hypertext is modelled as a finite state machine (*browsing automata*). Its states represent hypertext nodes, its transitions hypertext links. Local properties of nodes such as the availability of certain buttons, menu options, and content fragments are represented by Boolean state variables. The temporal logic HTL* (hypertext temporal logic), a syntactical extension of CTL* (computation tree logic)[6], is used as a specification language for

Regular Paper

global properties of browsing histories such as reachability of certain pages from other pages and availability of certain functions on pages. In [7] the method has been enhanced for the verification of quasi-parallel browsing activities in framesets.

An early attempt to guarantee the soundness and completeness of user manuals for technical systems has been described in [8, 9]. Consistency of manuals is defined as the close correspondence of the manual's content to a formal specification of the system's behaviour. In [8] an annotated finite state machine modelling the behaviour of a technical system is used as the basis to automatically construct a manual consistent with the specification. In [9] properties of the system and the corresponding manual are expressed in a *temporal logic of actions* and verified using Prolog[10].

The MMiSS-project[1] aims at providing a web-based adaptive educational system in the domain of Safe Systems. Consistency and completeness of dynamically assembled or frequently changed learning documents are a major concern of the project. Therefore, the semantical interrelationships of learning content are represented on a fine grained level based on ontologies of structural units and concepts used in the document[2]. A rule-based approach for the specification of constraints on semantical interrelationships is sketched but not defined in detail. Learning content, metadata which describe the content, and ontologies which describe the logical structure of the content are represented using a proprietarily enhanced LaTeX format. As a result, the integration of foreign web resources requires a considerable re-engineering effort.

In our work we extend the approach of Stotts and Furuta[5] in two ways: 1) towards the specification and verification of properties of the document's content and narrative structure rather than its hypertext or frameset structure, and 2) towards the adoption of terminological knowledge organized in ontologies to be able to reason about document fragments from different sources with potentially different content models and document formats.

3 Approach

3.1 Knowledge Representation

Fig.2 illustrates how knowledge about an integrated document is represented for consistency checking.

Assume there is a hypermedia document d available in *context* C_d which integrates the fragments e and f from foreign *contexts* C_e and C_f respectively (bottom of Fig.2). We say that a (part of a) document d belongs to a context C_d if it conforms to the document and metadata format as well as the style of metadata tagging of context C_d. All documents of a specific context are assumed to be homogeneous in terms of document and metadata formats while documents of different contexts might be implemented using different document formats and tagged according to different metadata models.

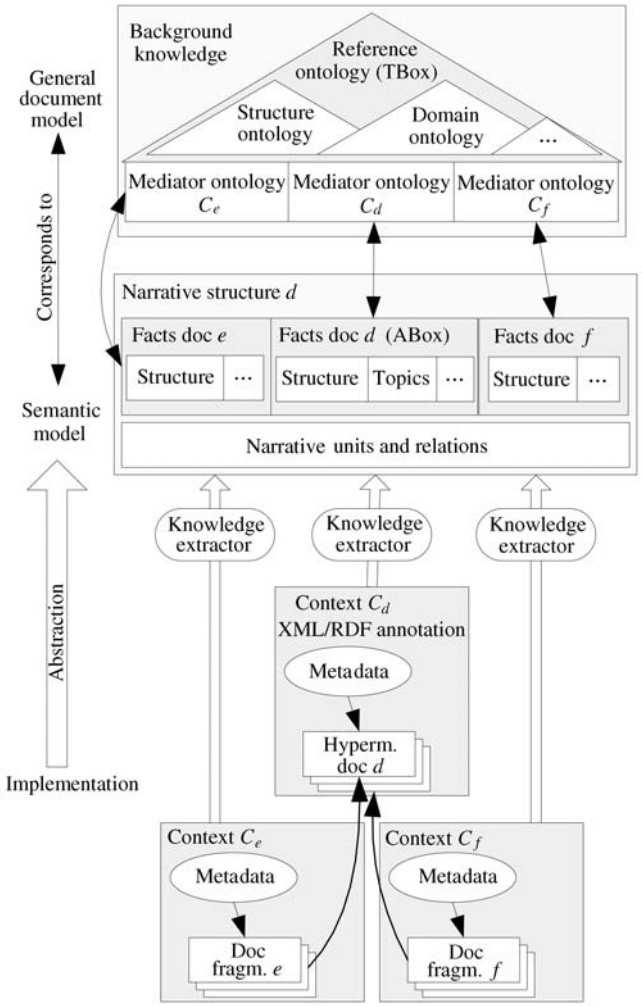

Fig.2. Knowledge representation of integrated hypermedia documents.

The first step is the abstraction from implementation details such as document and metadata format by representing the relevant knowledge about the structure and content of the document within a *semantic model* (Fig.2 left hand side). In the case of structured document and metadata formats e.g., [1, 11–13], *knowledge extractor* components (middle of Fig.2) select the relevant metadata and translate into description logic (DL) assertions for instance by mapping mechanisms as proposed in [14, 15]. In the case of unstructured document formats, information extraction tools[16–20] can help to obtain the required knowledge. In this article we do not further address these techniques but focus on representing and verifying semantical criteria on integrated documents.

The *semantic model* represents the *narrative structure* (middle of Fig.2) rather than the hypertext structure of the document. The narrative structure of a document determines recommended (but not necessarily enforced) flows of reading in standard situations. It guides the user through the content along coherent *narrative paths*. Complex web documents such as manuals, text

books, or web-based trainings usually offer alternative narrative paths for different target groups or information demands. Hence the narrative structure of a document is represented by a directed graph of *narrative units* (vertices) and *narrative relations* (edges) (middle of Fig.2 and Fig.3). Since it has to be guaranteed that the document makes sense along narrative paths the narrative structure is an appropriate basis for determining the semantical consistency of documents.

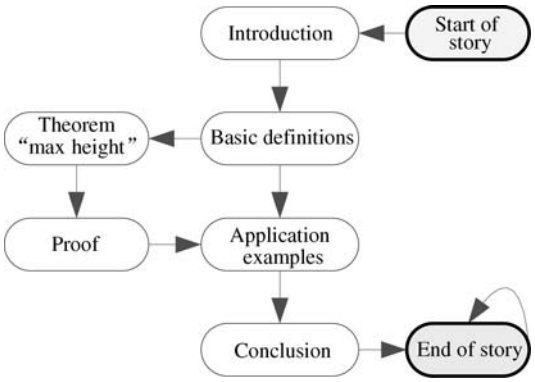

Fig.3. Narrative graph of a document having a short and an extended narrative path.

In addition, the narrative structure contains *facts* (middle of Fig.2) about the content *structure* (sections, definitions, examples, ...), the *topics*, and further aspects (e.g., educational objectives and prerequisites) of each narrative unit. Facts are represented in DL by a set of assertions (ABox).

The facts represented in the semantic model correspond to general *background knowledge* consisting of one or more ontologies represented by DL axioms (TBox) (top of Fig.2). The *structure* of the document is described in terms of a *structure ontology* which represents a general content model for a certain type of document (e.g., teachware[11]). The topics of the document are described in terms of a *domain ontology* representing entities and relationships among entities in the domain of discourse[1,21]. Dependent on the type of document other ontologies may be used to represent models of further aspects which might be relevant for determining the document's consistency such as learning objectives, prerequisites and processes[22−24].

Remark 1 (Ontologies). The aim of using various ontologies as sketched above is firstly to ensure interoperability in reasoning on documents which integrate fragments from different contexts and secondly to increase the expressiveness of specifications of semantical properties. However, ontologies are not an essential part of the framework: in easy (homogeneous) application scenarios useful properties can be expressed and checked without any additional background knowledge required.

For a base of documents an appropriate set of ontologies is selected as a *reference ontology* (top of Fig.2). As the document integrates fragments from different contexts the *facts* derived from the document format and metadata in the semantic model may not correspond exactly to the chosen reference ontology. For instance, the term *tree* may represent a plant in one context and a data structure in another. Therefore, for each context a *mediator ontology* (top of Fig.2) may be defined serving as an interface between the local information model of each context and the reference ontology.

In the following example we illustrate how ontologies help to integrate knowledge from different local presentations and to derive implicit knowledge.

Example 2 (Knowledge Representation). Assume a document d in context C_d contains a paragraph *defBTree* defining the term "BayerTree" and imports a fragment e from context C_e containing an example *exaBTree* of "BayerTree". From the XML syntax or external metadata certain *facts* about the document are derived and represented in the *semantic model* using description logics assertions[25]:

$$Paragraph(defBTree), \quad (1)$$
$$defines(defBTree,\ bayertree), \quad (2)$$
$$Example(exaBTree), \quad (3)$$
$$hasTopic(exaBTree, b\text{-}tree). \quad (4)$$

The assertions (1) and (3) describe the fragments *defBTree* and *exaBTree* in terms of the *structure ontology*. The assertions (2) and (4) describe the fragments in terms of the the *domain ontology*.

We assume that the *mediator ontology* of the context C_d contains the assertion $BTree(bayertree)$ representing the fact that *bayertree* is a *BTree*-topic in context C_d. Similarly the mediator ontology of context C_e contains the assertion $BTree(b\text{-}tree)$.

In the *structure ontology* further knowledge about the concepts *Paragraph* and *Example*, and the roles $defines$ and $hasTopic$ is represented, e.g.,

$$Definition \doteq Paragraph \sqcap \exists defines.Topic,$$
$$Example \doteq Paragraph \sqcap \exists exemplifies.Topic$$
$$exemplifies \sqsubseteq hasTopic,$$
$$defines \sqsubseteq hasTopic.$$

In the *domain ontology* further information about topics and relationships of topics is represented, e.g.,

$$BTree \sqsubseteq Tree,$$
$$BTree \sqsubseteq IndexStructure,$$
$$IndexStructure \sqsubseteq \exists usedIn.DataBases.$$

Let KB be a knowledge base containing the semantic model, the mediator ontologies, and the structure and domain ontology as sketched above. We now can reason in a uniform way about the two fragments and infer, for instance, that fragment *defBTree* is a *Definition* and that both fragments deal with something used in data bases:

$$KB \models (definition \sqcap \exists hasTopic.BTree)(defBTree),$$

$KB \models (\exists hasTopic.\exists usedIn.DataBases)(defBTree)$,
$KB \models (\exists hasTopic.\exists usedIn.DataBases)(exaBTree)$.

3.2 Representing Conventions and Criteria

For expressing semantical constraints on the narrative structure of documents, we enhance the branching time temporal logic CTL (computation tree logic[6]) to $\mathsf{CTL}_{\mathcal{DL}}$ in a way that concept descriptions instead of atomic propositions can be used for local constraints. This enables us to make use of terminological background knowledge represented in the reference and mediator ontologies for the specification and verification of semantical properties which in turn is particularly useful for handling integrated documents from heterogeneous sources. We formally define the syntax and semantics of $\mathsf{CTL}_{\mathcal{DL}}$ in Subsection 4.2. For now we briefly explain the intuitive meaning of the branching time modal connectives of $\mathsf{CTL}_{\mathcal{DL}}$, give some sample specifications, and state their intuitive meaning.

$\mathsf{CTL}_{\mathcal{DL}}$ is in general interpreted on the nodes (called *states*) and *paths* of a directed graph which represents a narrative structure of a web document in our application. The modal connectives used in the following examples are:

• Path quantifiers:

Ap: "all paths p" — in *all* paths starting from the current state p holds;

Ep: "some path p" — on *some* path starting from the current state p holds;

• Tense operators:

Fp: "eventually/future p" — on the current path p holds at least once from now on;

Gp: "always/globally p" — on the current path p holds in every state from now on;

Xp: "next p" — on the current path p holds in the *next* state.

Example 3 ($\mathsf{CTL}_{\mathcal{DL}}$ Specifications on Narrative Structures). On every narrative path in the document the reader will learn something about index structures:

$$\mathsf{AF}\ \exists hasTopic.IndexStructure.$$

There is at least one narrative path which contains an application of b-trees in the context of data bases:

$$\mathsf{EF}\ \exists hasTopic.DataBases \sqcap \exists applies.BTree.$$

Whenever a fragment is formal or discusses a central topic an illustration is provided in the sequel:

$$\mathsf{AG}(Formal \sqcup \exists hasTopic.CentralTopic \Rightarrow Illustration \vee \mathsf{AX}\,Illustration).$$

The document starts with an introduction followed by some presentations, each offering an optional exercise with at least two tasks and ending with a summary:

$$Intro \wedge \mathsf{AF}\ Presentation \wedge \mathsf{AG}(Presentation \Rightarrow \mathsf{EX}\,Exercise \sqcap \exists^{\geqslant 2} contains.Task \wedge \mathsf{AF}\ Summary).$$

4 Formal Framework

4.1 Representation of Knowledge about Documents

In the sequel we formally define the *narrative structure* of a document. The narrative structure represents relevant aspects of the document's content and structure in a semantic model.

Definition 4 (Narrative Structure). *A narrative structure is a tuple* $NS = (NU, UID, m, u_0, u_e, NR, \boldsymbol{A}, cm)$ *where*

• NU *is a nonempty finite set of document fragments each constituting a complete narrative unit*;

• UID *is a set of individuals such that* $\#UID = \#NU$;

• $m : UID \rightarrow NU$ *is a bijective function mapping individuals onto narrative units*;

• $u_0 \in UID$ *is the representation of a start unit of all narrative paths*;

• $u_e \in UID$ *is the representation of an end unit of all narrative paths*;

• $NR \subseteq UID \times UID$ *is a representation of narrative relations between the narrative units of a document such that* $(u_e, u_e) \in NR$, *and* $\forall u \in UID\ \exists(u_0, \ldots, u, \ldots, u_e) \in \{(u_0, u_1, \ldots, u_n) | \forall i \in \{0, \ldots, n-1\} : (u_i, u_{i+1}) \in NR\}$, *i.e., for each narrative unit* $m(u)$ *there is a narrative path leading from* $m(u_0)$ *over* $m(u)$ *to* $m(u_e)$. $(u, v) \in NR$ *expresses that unit* $m(v)$ *should not be read before unit* $m(u)$;

• $\boldsymbol{A} = \{A_u | u \in UID\}$ *is a set of ABox-Assertions of a description logic* $\mathcal{DL}$. *Each* $A_u \in \boldsymbol{A}$ *contains (local) assertions about the content of a given narrative unit* $m(u)$. *All ABoxes are assumed to be finite*;

• cm: $UID \rightarrow Ctxt$ *is a function mapping each narrative unit* $u \in UID$ *onto the source context* $cm(u)$ *the narrative unit* u *is received from. Ctxt denotes the set of source contexts of narrative units.*

The individuals in UID represent actual document fragments NU. We have defined UID to represent exactly the narrative units of a given document. We also assume to have knowledge about the concrete mapping m of individuals onto document units. Moreover, we assume that each narrative unit u belongs in its entirety to a single source context $cm(u)$ and is implemented and described by metadata according to the respective knowledge representation models and data formats of context $cm(u)$.

We further assume a document to have a distinct start and end unit u_0 and u_e. The start unit is the entry point of the document (associated with the document's URL). The end unit is the fragment from where the document is left. u_0 and u_e may represent "pseudo fragments" which do not have any real content. We

require the narrative structure of a document to be coherent in the sense that every narrative unit is included in some narrative path from u_0 to u_e. Non-coherent narrative structures can be broken down into coherent parts which are treated separately. For technical reasons we require $(u_e, u_e) \in NR$ such that NR is a left total relation, i.e., $\forall u \in UID\ \exists v \in UID : (u, v) \in NR$.

Example 5 (Narrative Structure). Fig.4 illustrates a narrative structure containing an introduction, the definition, and an example of the data structure *tree*, an optional illustration of the term *tree*, and a conclusion. It is represented by

- $UID = \{intro, illuTree, defTree, exaTree, concl\}$;
- $u_0 = intro$;
- $u_e = concl$;
- $NR = \{(intro, illuTree), (intro, defTree), (illuTree, defTree), (defTree, exaTree), (exaTree, concl), (concl, concl)\}$
- $cm = UID \to Ctxt$: $cm(intro) = cm(defTree) = cm(concl) = m$, $cm(illuTree) = f_1$, $cm(exaTree) = f_2$;
- $\boldsymbol{A} = \{A_{intro}, A_{illuTree}, A_{defTree}, A_{exaTree}, A_{concl}\}$

where

$$A_{intro} = \{Introduction(intro)\},$$
$$A_{illuTree} = \{Illustration(illuTree), hasTopic(illuTree, tree)\},$$
$$A_{defTree} = \{Definition(defTree), hasTopic(defTree, tree)\},$$
$$A_{exaTree} = \{exemplifies(exaTree, tree)\},$$
$$A_{concl} = \{Conclusion(concl)\}.$$

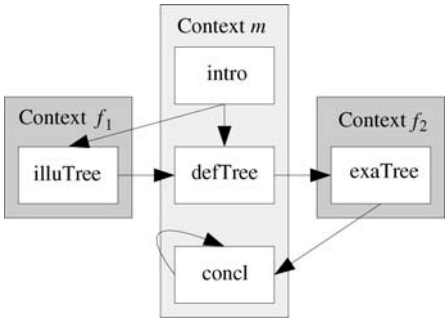

Fig.4. Narrative structure of a document across the contexts f_1, f_2 (foreign) and m (main).

4.2 Specification of Semantical Properties by $\mathsf{CTL}_{\mathcal{DL}}$

We now define formally the syntax and semantics of the branching time temporal logic $\mathsf{CTL}_{\mathcal{DL}}$ which we suggest as a formalism for expressing semantical constraints on documents.

Definition 6 (Syntax of $\mathsf{CTL}_{\mathcal{DL}}$). *Let $\mathcal{DL}$ be a decidable description logic allowing for assertions on individuals (ABoxes) such as $\mathcal{ALC}$[25] or $\mathcal{SHIQ}$[26]. Let $\boldsymbol{C}_{\mathcal{DL}}$ be the set of concept descriptions as defined by the syntax of $\mathcal{DL}$. The set of $\mathsf{CTL}_{\mathcal{DL}}$ formulae is the set of state formulae generated by the following rules:*

(S1) *each concept description $C \in \boldsymbol{C}_{\mathcal{DL}}$ is a state formula*;

(S2) *if p, q are state formulae then so are $p \wedge q$ and $\neg_f p$*;

(P0) *if p, q are state formulae then $\mathsf{X}p$ and $p\ \mathsf{U}\ q$ are path formulae*;

(S3) *if p is a path formula then $\mathsf{E}p$ and $\mathsf{A}p$ are state formulae.*

A $\mathsf{CTL}_{\mathcal{DL}}$ formula is interpreted on $\mathcal{DL}$ knowledge bases and individuals "temporally" structured in the following way:

Definition 7 ($\mathsf{CTL}_{\mathcal{DL}}$ Temporal Structure). *A $\mathsf{CTL}_{\mathcal{DL}}$ temporal structure is a tuple $M = (S, R, \boldsymbol{KB})$ where*

- *S is a nonempty set of $\mathcal{DL}$ individuals representing states*;
- *$R \subseteq S \times S$ is a left total binary relation on S*;
- *$\boldsymbol{KB} = \{KB_s \mid s \in S\}$ is a set of consistent $\mathcal{DL}$ knowledge bases, i.e., each $KB_s \in \boldsymbol{KB}$ has a model $\mathcal{I}_s \models KB_s$. Note that for $s, t \in S : s \neq t$ does not imply $KB_s \neq KB_t$ (see example below).*

Example 8 (Temporal Structure). Let $M = (S, R, \boldsymbol{KB})$ where

- $S = \{intro, exaTree, concl\}$;
- $R = \{(intro, exaTree), (intro, concl), (exaTree, concl), (concl, concl)\}$;
- $\boldsymbol{KB} = \{KB_{intro}, KB_{exaTree}, KB_{concl}\}$ such that $KB_{intro} = KB_{exaTree} = KB_{concl} =$ $\{BinTree \sqsubseteq Tree,$ $exemplifies \sqsubseteq hasTopic,$ $Introduction(intro),$ $exemplifies(exaTree, aTree),$ $BinTree(aTree), Conclusion(concl)\}$.

Then the truth of $\mathsf{CTL}_{\mathcal{DL}}$ formulae, e.g., $p = \mathsf{AG}(Introduction \Rightarrow \mathsf{AF}\, Conclusion)$ and $q = \mathsf{AF}\exists hasTopic.Tree$ can be evaluated w.r.t. M (see Example 11).

The *semantics* of $\mathsf{CTL}_{\mathcal{DL}}$ defines when a formula $p \in \mathsf{CTL}_{\mathcal{DL}}$ is *true* in a structure $M = (S, R, \boldsymbol{KB})$ at state $s_0 \in S$, in symbols: $M, s_0 \models p$. For the definition of $\models$ we define a *fullpath* in S as a sequence $x = (s_0, s_1, s_2, \ldots)$ such that $\forall i \in \mathbb{N}_0 : (s_i, s_{i+1}) \in R$. FP_{s_0} denotes the set of fullpaths in S starting from state s_0: $FP_{s_0} = \{(s_0, s_1, \ldots) | \forall i \in \mathbb{N}_0 : (s_i, s_{i+1}) \in R\}$.

$M, x \models p$ denotes that a path formula p is *true* in the structure M for the fullpath x.

Definition 9 (Semantics of $\mathsf{CTL}_{\mathcal{DL}}$). *Let $\models_{\mathcal{DL}}$ be the entailment operator of $\mathcal{DL}$ as defined in* [27]. *Let $M = (S, R, \boldsymbol{KB})$ be a temporal structure (Definition 7), $s_0 \in S$ a state, and $x = (s_0, s_1, \ldots)$ a fullpath in FP_{s_0}. We define $\models$ inductively as follows*:

(S1) *for $C \in \boldsymbol{C}_{\mathcal{DL}}$: $M, s_0 \models C$ iff $KB_{s_0} \models_{\mathcal{DL}} C(s_0)$*;

(S2) *$M, s_0 \models p \wedge q$ iff $M, s_0 \models p$ and $M, s_0 \models q$, $M, s_0 \models \neg_f p$ iff not $M, s_0 \models p$*;

(P0) *$M, x \models \mathsf{X}p$ iff $M, s_1 \models p$, $M, x \models p\ \mathsf{U}\ q$ iff $\exists i \in \mathbb{N}_0$ $[M, s_i \models q$ and $\forall j \in \mathbb{N}_0$ $(j < i$ implies $M, s_j \models p)]$*;

(S3) *$M, s_0 \models \mathsf{E}p$ iff $\exists x \in FP_{s_0}$ such that $M, x \models p$,*

$M, s_0 \models \mathsf{A}p$ *iff* $\forall x \in FP_{s_0}$: $M, x \models p$.

Remark 10 (Syntax and Semantics of $\mathsf{CTL}_{\mathcal{DL}}$). Some comments on the definitions above are in order:

• In rules (S2) above we denote the negation of a state formula $\neg_f p$ with an index to distinguish it from the negation used in $\mathcal{DL}$ concept descriptions. In fact, the two negations have a different semantics: for a $\mathcal{DL}$ concept C the state formula $\neg C$ holds in M, s_0 iff for all models $\mathcal{I}$ of KB_{s_0} it can be proven that $s_0^{\mathcal{I}} \notin C^{\mathcal{I}}$. In contrast, $\neg_f C$ holds in M, s_0 iff any proof $KB_{s_0} \models C(s_0)$ *fails*, i.e., it cannot be proven that $s_0^{\mathcal{I}} \in C^{\mathcal{I}}$ for all models $\mathcal{I}$ of KB_{s_0}.

Obviously it holds:

$$M, s_0 \models \neg C \Rightarrow M, s_0 \models \neg_f C \tag{5}$$

but

$$M, s_0 \models \neg_f C \not\Rightarrow M, s_0 \models \neg C. \tag{6}$$

• We use the following abbreviations as defined in [6]:

$$p \vee q = \neg_f(\neg_f p \wedge \neg_f q)$$
$$p \Rightarrow q = \neg_f p \vee q$$
$$\mathit{true} = \neg_f p \vee p,\ \mathit{false} = \neg_f\ \mathit{true}$$
$$\mathsf{F}p = \mathit{true}\ \mathsf{U}\ p\ (\text{“eventually } p\text{”})$$
$$\mathsf{G}p = \neg_f \mathsf{F} \neg_f p\ (\text{“always } p\text{”})$$
$$p\ \mathsf{B}\ q = \neg_f((\neg_f p)\ \mathsf{U}\ q)\ (\text{“}p \text{ before } q\text{”}).$$

• The binding precedence of connectives is: the connectives of the sublanguage $\mathcal{DL}$ ($\forall, \exists, \neg, \sqcap, \sqcup$) have highest binding power, followed by the path operators A, E, temporal operators $\mathsf{F}, \mathsf{G}, \mathsf{X}, \mathsf{U}, \mathsf{B}$, and finally $\neg_f, \wedge, \vee, \Rightarrow$.

• In contrast to existing temporally extended description logics[28–30], $\mathsf{CTL}_{\mathcal{DL}}$ has a rather modular structure: a $\mathsf{CTL}_{\mathcal{DL}}$ formula p is a second order formula over a set of $\mathcal{DL}$ knowledge bases $\boldsymbol{KB}$. More precisely, p is interpreted on reified individuals themselves representing objects of a domain which in general is not known completely (open world assumption). In our application we assume a certain set of individuals *ID* and relationships *NR* to be known completely such that the more efficient closed world reasoning (model checking) can be applied for that part of knowledge. The modular structure of $\mathsf{CTL}_{\mathcal{DL}}$ serves well for integrating open and closed world reasoning within one framework and enables checking global properties across distributed knowledge bases as we will see later on.

Example 11 (Semantics of $\mathsf{CTL}_{\mathcal{DL}}$). Let M, p, and q be as in Example 8, i.e, $p = \mathsf{AG}(\mathit{Introduction} \Rightarrow \mathsf{AF}\ \mathit{Conclusion})$ and $q = \mathsf{AF}\exists\ \mathit{hasTopic.Tree}$. Then

$$M, \mathit{intro} \models p$$
$$M, \mathit{exaTree} \models p$$
$$M, \mathit{concl} \models p$$
$$M, \mathit{intro} \not\models q$$
$$M, \mathit{exaTree} \models q$$
$$M, \mathit{concl} \not\models q$$
$$M, \mathit{intro} \not\models \neg\mathit{Conclusion}$$
$$M, \mathit{intro} \models \neg_f \mathit{Conclusion}$$
$$M, \mathit{exaTree} \models \neg\forall \mathit{hasTopic.}\neg \mathit{Tree}$$
$$M, \mathit{exaTree} \models \neg_f\forall \mathit{hasTopic.}\neg \mathit{Tree}.$$

Proposition 12 (Relationship Between $\mathsf{CTL}_{\mathcal{DL}}$ and CTL). *Let $\models_{\mathsf{CTL}_{\mathcal{DL}}}$ be the entailment operator as defined in Definition 9 and $\models_{\mathsf{CTL}}$ the entailment operator defined by the standard* CTL *semantics*[6]. *Every $\mathsf{CTL}_{\mathcal{DL}}$ formula p, interpreted over $(S, R, \boldsymbol{KB})$, can be interpreted as a* CTL *formula over a structure (S, R, L) such that $(S, R, \boldsymbol{KB}), s_0 \models_{\mathsf{CTL}_{\mathcal{DL}}} p \Leftrightarrow (S, R, L), s_0 \models_{\mathsf{CTL}} p$.*

Proof. Let p be a $\mathsf{CTL}_{\mathcal{DL}}$ formula. Let $M = (S, R, \boldsymbol{KB})$ be a structure as defined in Definition 7. Let $\boldsymbol{C}_p$ denote the set of concept descriptions used in p. Let L be a labelling

$$L : S \to \mathcal{P}(\boldsymbol{C}_p) \text{ with}$$
$$L(s) := \{C \in \boldsymbol{C}_p \mid KB_s \models_{\mathcal{DL}} C(s)\}.$$

Such a labelling exists because we assumed $\mathcal{DL}$ to be decidable.

By interpreting all concept descriptions $C \in \boldsymbol{C}_p$ as atomic propositions we can now interpret p in terms of standard CTL semantics over the structure $M = (S, R, L)$. The semantics of CTL is identical to the semantics of $\mathsf{CTL}_{\mathcal{DL}}$ except for rule (S1) which in the case of CTL reads:

(S1′) $M, s_0 \models C$ iff $C \in L(s_0)$.

Let $C \in \boldsymbol{C}_p$ be a concept description in p. Then

$$\begin{aligned} & (S, R, \boldsymbol{KB}), s_0 \models_{\mathsf{CTL}_{\mathcal{DL}}} C \\ \Leftrightarrow\ & KB_{s_0} \models_{\mathcal{DL}} C(s_0) && (\text{def. } \models_{\mathsf{CTL}_{\mathcal{DL}}}) \\ \Leftrightarrow\ & C \in L(s_0) && (\text{def. } L) \\ \Leftrightarrow\ & (S, R, L), s_0 \models_{\mathsf{CTL}} C && (\text{def. } \models_{\mathsf{CTL}}) \end{aligned}$$

Since $\models_{\mathsf{CTL}_{\mathcal{DL}}}$ and $\models_{\mathsf{CTL}}$ coincide in the cases (S2, S3, P0) the equivalence $(S, R, \boldsymbol{KB}), s_0 \models_{\mathsf{CTL}_{\mathcal{DL}}} p \Leftrightarrow (S, R, L), s_0 \models_{\mathsf{CTL}} p$ follows by induction on the structure of p. □

Relevant for our application is the model checking problem of $\mathsf{CTL}_{\mathcal{DL}}$:

Definition 13 (Model Checking Problem of $\mathsf{CTL}_{\mathcal{DL}}$). *Let M be a finite structure $(S, R, \boldsymbol{KB})$ as defined in Definition 9 and let $p \in \mathsf{CTL}_{\mathcal{DL}}$ be a formula. The model checking problem of $\mathsf{CTL}_{\mathcal{DL}}$ is to decide for all $s \in S$, if $M, s \models p$.*

Proposition 14 (Complexity of the Model Checking Problem). *The model checking problem for $\mathsf{CTL}_{\mathcal{AL}}$ is* ExpTime*-hard.*

Proof. Since model checking a $\mathsf{CTL}_{\mathcal{AL}}$ formula involves deciding the logical implication $KB_s \models_{\mathcal{AL}} C(s)$, and logical implication w.r.t. a knowledge base with general concept inclusion axioms is ExpTime-hard for $\mathcal{AL}$ (see [31]), the model checking problem for $\mathsf{CTL}_{\mathcal{AL}}$ is at least ExpTime-hard.

Proposition 12 implies that the model checking problem of $\mathsf{CTL}_{\mathcal{AL}}$ w.r.t. a structure $(S, R, \boldsymbol{KB})$ is equivalent to the CTL model checking problem w.r.t. a structure (S, R, L). Since S and p are assumed to be finite, $(S, R, \boldsymbol{KB})$ can be transformed into a structure (S, R, L) in polynomial steps (see proof of Proposition 12). In each step the logical implication of an assertion w.r.t. an $\mathcal{AL}$ knowledge base must be decided which is known to be ExpTime-hard[31]. Hence the transformation of $(S, R, \boldsymbol{KB})$ into (S, R, L) is ExpTime-hard. The model checking problem of CTL is known to be in deterministic polynomial time[6]. Thus the model checking problem for $\mathsf{CTL}_{\mathcal{AL}}$ can be reduced to a polynomial problem in ExpTime. As a result the model checking problem of $\mathsf{CTL}_{\mathcal{AL}}$ is at most ExpTime-hard. □

Remark 15. For languages $\mathcal{DL}$ at least as expressive as $\mathcal{AL}$ the complexity of the model checking problem for $\mathsf{CTL}_{\mathcal{DL}}$ is determined by the complexity of deciding logical implication in $\mathcal{DL}$.

5 Verification of Narrative Structures

5.1 Definition of the Verification Problem

The semantical consistency of a narrative structure *NS* is determined w.r.t. a base of background knowledge B defined as follows.

Definition 16 (Background Knowledge). *A base of background knowledge B is a tuple $(RO, \boldsymbol{MO})$ where*

- *RO (reference ontology) is a $\mathcal{DL}$ knowledge base representing a commonly agreed upon reference model for content structure, discourse domain, learning objectives, and other aspects necessary to describe a document.*
- *$\boldsymbol{MO} = \{MO_c | c \in Ctxt\}$ (mediator ontologies) is a set of $\mathcal{DL}$ knowledge bases each serving as a mediator between the local knowledge representation model of a context c and RO ($Ctxt$ denotes the set of contexts). MO_c may be empty in the case that the extracted facts about the narrative units of a context correspond exactly to the reference ontology.*

The verification problem for a narrative structure *NS*, a base of background knowledge B, and a $\mathsf{CTL}_{\mathcal{DL}}$ formula p can be broken down into two parts:

- *metadata consistency*: checking *local* and *global consistency* of *NS* with B (Definition 17);
- *narrative consistency*: checking *satisfaction* of p in *NS* w.r.t. B. This requires that *NS* is locally consistent with B. However, *NS* does not need to be *globally* consistent with B.

Definition 17 (Local/Global Consistency). *Let $NS = (NU, UID, m, u_0, u_e, NR, \boldsymbol{A}, cm)$ be a narrative structure and $B = (RO, \boldsymbol{MO})$ background knowledge. Recall that $\boldsymbol{A} = \{A_u | u \in UID\}$ is a set of $\mathcal{DL}$ ABoxes such that for each $u \in UID$ A_u contains a set of $\mathcal{DL}$ assertions representing extracted facts about the narrative unit $m(u)$ (Definition 4). Let KB_u denote the local knowledge base w.r.t. a narrative unit $m(u)$:*
for $u \in UID$: $KB_u = RO \cup MO_{cm(u)} \cup A_u$. NS is locally consistent with B iff each local knowledge base KB_u is satisfiable:

$$\forall u \in UID \exists \mathcal{I} : \mathcal{I} \models KB_u,$$

i.e., each local knowledge base KB_u has a model $\mathcal{I} \models KB_u$.

NS is globally consistent with B iff the union of all local knowledge bases is satisfiable:

$$\exists \mathcal{I} : \mathcal{I} \models \bigcup_{u \in UID} KB_u.$$

Local consistency means that the knowledge about each narrative unit in ABox A_u conforms to the reference ontology RO and respective mediating ontology $MO_{cm(u)}$. Global consistency means that the entire knowledge about the document $\boldsymbol{A}$ conforms to the entire background knowledge B. In the case of integrated documents using resources from distributed, autonomous repositories global consistency of knowledge and knowledge representation schemata is usually quite hard to maintain[32].

Example 18 (Local/Global Consistency). Let *NS* be the narrative structure of Example 5. Let RO be a $\mathcal{DL}$ TBox as follows:

$$Plant \sqcap Datastructure \doteq \bot \tag{7}$$
$$NatTree \sqsubseteq Plant \sqcap Tree \tag{8}$$
$$DSTree \sqsubseteq Datastructure \sqcap Tree \tag{9}$$
$$BinTree \sqsubseteq DSTree \tag{10}$$
$$\ldots$$

Let $MO_m = \{DSTree(tree)\}$ (11)
$$MO_{f_1} = \{NatTree(tree)\} \tag{12}$$
$$MO_{f_2} = \{BinTree(tree),\ exemplifies \sqsubseteq hasTopic,\ \exists exemplifies.\top \doteq Example\}. \tag{13}$$

Then *NS* is locally but not globally consistent with B, because $RO \cup MO_{cm(u)} \cup A_u$ is satisfiable for each $u \in UID$ and $RO \cup MO_{f_1} \cup MO_m$ is inconsistent since (7) in RO prevents *tree* from being an instance of *DSTree* in (11) *and NatTree* in (12). As a result $\bigcup_{u \in UID}(RO \cup MO_{cm(u)} \cup A_u)$ is not satisfiable.

The second verification problem for documents as proposed in this paper is to decide if a given narrative structure *NS* satisfies an $\mathsf{CTL}_{\mathcal{DL}}$ formula p under a base of ontological background knowledge B.

5.2 Checking Narrative Consistency

Let $B = (RO, \boldsymbol{MO})$ be a base of background knowledge as defined in Definition 16. Let $NS = (NU, UID, m, u_0, u_e, NR, \boldsymbol{A}, cm)$ be a narrative structure (see Definition 4) of a web document such that *NS* is locally consistent with B. Let as before KB_u denote the local knowledge base w.r.t. a narrative unit $m(u)$:
for $u \in UID, KB_u := RO \cup MO_{cm(u)} \cup A_u$.

Let p be a $\mathsf{CTL}_{\mathcal{DL}}$ formula. We define the verification problem of NS, p, B as follows.

Definition 19 (Satisfaction of a Specification). *A narrative structure NS satisfies p under B (in symbols* $NS \models_B p$*) iff* $(UID, NR, \{KB_u | u \in UID\}), u_0 \models p$.

Remark 20. Since for a given narrative structure NS $(UID, NR, \{KB_u | u \in UID\})$ is finite and NR is left total, the verification problem of NS, p, B is a model checking problem restricted to state u_0.

Example 21 (Satisfaction of a Specification). Let NS be as in Example 5 and B be as in Example 18. Then

$$NS \models_B \mathsf{E}(\mathit{Illustration} \sqcap \exists\, \mathit{hasTopic}.\mathit{Tree}\ \mathsf{B}\ \mathit{Definition} \sqcap \exists\, \mathit{hasTopic}.\mathit{Tree}).$$

There is some narrative path in NS — the path (*intro, illuTree, defTree*, ...) — such that an *Illustration* of a *Tree* is presented before a *Definition* of a *Tree*. However,

$$NS \not\models_B \mathsf{E}(\mathit{Illustration} \sqcap \exists\, \mathit{hasTopic}.\mathit{Datastructure}\ \mathsf{B}\ \mathit{Definition} \sqcap \exists\, \mathit{hasTopic}.\mathit{Datastructure}),$$

since *tree*, the topic of the *Illustration illuTree*, is not a *Datastructure* in state *illuTree*:

$$KB_{\mathit{illuTree}} \not\models (\exists \mathit{hasTopic}.\mathit{Datastructure})(\mathit{illuTree}).$$

5.3 Abstract Algorithm for Model Checking $\mathsf{CTL}_{\mathcal{DL}}$

Given decision procedures for instance checking w.r.t. a $\mathcal{DL}$ knowledge base ($KB \vdash_{\mathcal{DL}} C(a)$) and for model checking a CTL formula p ($M, s_0 \vdash_{\mathsf{CTL}} p$) a straight forward algorithm for the verification of a narrative structure $NS = (NU, UID, m, u_0, u_e, NR, \boldsymbol{A}, cm)$ against a $\mathsf{CTL}_{\mathcal{DL}}$ formula p w.r.t. a base of background knowledge $B = (RO, \boldsymbol{MO})$ is:

Algorithm 1. Verification of the Narrative Consistency of a Document by $\mathsf{CTL}_{\mathcal{DL}}$ Model Checking

function verify($UID, NR, \boldsymbol{A}, cm, RO, \boldsymbol{MO}, u_0, p$):
  Boolean {
    $\boldsymbol{C}_p := \{C \mid C$ concept description in $p\}$
  for each $u \in UID$ { *#construct a labelling L*
    $L[u] := \{\}$
    for each $C \in \boldsymbol{C}_p$
      if $RO \cup MO_{cm(u)} \cup A_u \vdash_{\mathcal{DL}} C(u)$ **then**
        $L[u] := L[u] \cup \{C\}$
  }
  return $(UID, NR, L), u_0 \vdash_{\boldsymbol{CTL}} p$
}

Proposition 12 and its proof imply that *verify* is sound and complete given that $\vdash_{\mathcal{DL}}$ and $\vdash_{\mathsf{CTL}}$ are sound and complete. Let $n = \#UID$ be the number of fragments in NS and $m = \#\boldsymbol{C}_p$ the number of concept descriptions in p. The time complexity of *verify* can be calculated as $n \cdot m \cdot \mathit{Time}(\vdash_{\mathcal{DL}}) + \mathit{Time}(\vdash_{\mathsf{CTL}})$. In the case of $\mathcal{DL} = \mathcal{AL}$ there is a procedure $\vdash_{\mathcal{AL}} \in$ ExpTime and a procedure $\vdash_{\mathsf{CTL}} \in \boldsymbol{P}$ such that *verify* $\in$ ExpTime which is optimal (refer to Proposition 14).

Remark 22 (Time Complexity). The following properties of the *verify* algorithm can efficiently reduce the computational costs in our application.

- Deciding on properties of local states is separable from model checking a global structure. Hence the labelling of states in the *verify* algorithm can be calculated offline and in parallel for a given set of document fragments, background knowledge and specifications. If the labelling of states is already known the verification of a document can be done in polynomial time.
- We can benefit from the monotonic behaviour of reasoning in description logics: previously computed labellings of fragments keep valid when new knowledge or new document fragments become available. This is important if we think of large, potentially distributed, repositories and knowledge bases.

6 Analysis of the Relationship of $\mathsf{CTL}_{\mathcal{DL}}$ and Other Temporal Description Logics

There are a large variety of approaches to temporally extended description logics[28,29,33–35]. Among them we focus on the branching time logic $\mathcal{DPCTL}^*$[34] as a representative for illustrating the differences of $\mathsf{CTL}_{\mathcal{DL}}$ with other temporal description logics. Its structure is most similar to $\mathsf{CTL}_{\mathcal{DL}}$ and the majority of all decidable point based temporal description logics are fragments of $\mathcal{DPCTL}^*$.

$\mathcal{DPCTL}^*$ is based on the standard description logic $\mathcal{ALC}$ with the following extensions:

- past tense operators S (since), $\Diamond_P$ (some time in the past), $\Box_P$ (always in the past);
- future tense operator U (until), $\bigcirc$ (next time), $\Diamond_F$ (some time in the future), $\Box_F$ (always in the future);
- path quantifiers A (all paths) and E (some path).

$\mathcal{DPCTL}^*$ is interpreted over ω-trees, i.e., trees the full branches of which are order isomorphic to $\langle \mathbb{N}_0, < \rangle$[30]. A $\mathcal{DPCTL}^*$ model $\mathfrak{M} = \langle \mathfrak{F}, H, D, I \rangle$ is composed of

- an ω-tree $\mathfrak{F}$ representing states (or moments in time) and a "proceed" relation between them;
- a set of full branches H in $\mathfrak{F}$ representing linearly ordered flows of time in $\mathfrak{F}$;
- a domain of objects D;
- an interpretation function I assigning each moment in time s an $\mathcal{ALC}$-interpretation $I(s)$ of $\mathcal{DPCTL}^*$ atomic concepts, roles, and constants.

For the complete definition of the syntax and semantics of $\mathcal{DPCTL}^*$ we refer the readers to [34]. As there are few restrictions on how path quantifies, temporal, and non-temporal connectives can be combined the expressiveness of $\mathcal{DPCTL}^*$ is very high and $\mathcal{DPCTL}^*$ can be considered as an "upper bound" of temporal description logics[34].

In contrast to $\mathcal{DPCTL}^*$, $\mathsf{CTL}_{\mathcal{DL}}$ formulae are interpreted on temporal structures $M = (S, R, \boldsymbol{KB})$ where $\boldsymbol{KB}$ is a set of $\mathcal{DL}$ knowledge bases, i.e., ontological background knowledge represented in one or more $\mathcal{DL}$

knowledge bases is adopted when determining the truth of a $\mathsf{CTL}_{\mathcal{DL}}$ formula. This is very useful for expressing properties on heterogeneous structures as illustrated in Examples 2 and 21. The ability to incorporate $\mathcal{DL}$ knowledge bases and inferences does not only distinguish $\mathsf{CTL}_{\mathcal{DL}}$ from $\mathcal{DPCTL}^*$ but also, to the best of our knowledge, from all other approaches to temporally extended description logics.

One might wonder whether either of the logics $\mathsf{CTL}_{\mathcal{DL}}$ and $\mathcal{DPCTL}^*$ contains the other one, i.e., whether for every $\mathsf{CTL}_{\mathcal{DL}}$ formula there is an equivalent $\mathcal{DPCTL}^*$ formula or vice versa. While $\mathcal{DPCTL}^*$ and $\mathsf{CTL}_{\mathcal{DL}}$ share many of their basic syntactical constructs their semantics are quite different. For a formal comparison of the expressive power of $\mathsf{CTL}_{\mathcal{DL}}$ and $\mathcal{DPCTL}^*$ we would have to clarify first what $\mathcal{DPCTL}^*$ expressions mean in terms of $\mathsf{CTL}_{\mathcal{DL}}$ models or alternatively what $\mathsf{CTL}_{\mathcal{DL}}$ formulae mean in terms of ω-tree models. This in turn requires a complete redefinition of either logic making the comparison results rather meaningless.

Instead of a formal comparison we examine the expressive power and limitations of $\mathcal{DPCTL}^*$ and $\mathsf{CTL}_{\mathcal{DL}}$ on examples related to our use case. An important question is whether we can represent document properties expressible in $\mathsf{CTL}_{\mathcal{DL}}$ equally well in terms of $\mathcal{DPCTL}^*$. This is not the case as following examples demonstrate.

Example 23 (Potential Local Properties). On all paths through the document the user is eventually presented something *potentially* new.

$$\mathsf{AF}\neg_f\forall teaches.known.$$

The precise semantics of the formula above is: on all paths we will eventually reach a narrative unit of the document for which *it cannot be proven* that all topics taught are known topics, i.e., *it is possible* that at least one of the topics taught is not known by the user. In contrast,

$$\mathsf{AF}\neg\forall teaches.known$$

expresses that on all paths eventually *there is provably one topic* which is not known.

The distinction between provable and potential properties within a formula requires second order expressions such as $\neg_f$ which are not available in $\mathcal{DPCTL}^*$. This distinction is desirable in our use case since the represented knowledge about documents has often to be assumed incomplete. Hence the situation that some property cannot be proven true because of incomplete knowledge should be addressable in a formalism for document properties.

The modularity of $\mathsf{CTL}_{\mathcal{DL}}$ allows for using very expressive description logics such as $\mathcal{SHOQ}(\boldsymbol{D})$ for local subexpressions. $\mathcal{SHOQ}(\boldsymbol{D})$ is relevant in our case since knowledge representation standards of the semantic web such as OWL-DL[36] are based on $\mathcal{SHOQ}(\boldsymbol{D})$. It is guaranteed that the resulting temporal logic is decidable as long as the non-temporal part of the logic is decidable. Hence $\mathsf{CTL}_{\mathcal{DL}}$ can be easily adapted to new description logics as they become available. This is not the case for general temporal description logics such as $\mathcal{DPCTL}^*$ since a straight forward extension of these logics easily results in undecidability[29]. The following $\mathsf{CTL}_{\mathcal{SHOQ}(\boldsymbol{D})}$ sentences are not expressible in $\mathcal{DPCTL}^*$.

Example 24 (High Local Expressiveness).

• On all narrative paths the Dijkstra search algorithm is presented:

$$\mathsf{AF}\exists presents.(SearchAlg \sqcap \exists invented By.\{Dijkstra\}).$$

Since $\mathcal{DPCTL}^*$ does not contain expressions for nominals, the subexpression $\{Dijkstra\}$ cannot be expressed in $\mathcal{DPCTL}^*$.

Furthermore, as opposed to $\mathcal{DPCTL}^*$, $\mathsf{CTL}_{\mathcal{SHOQ}(\boldsymbol{D})}$ allows for concrete domain expressions which are required, e.g., for expressing numerical conditions on document fragments.

• All learning units should take between 30 and 60 minutes learning time:

$$\mathsf{AG}(learning Unit \rightarrow \forall has Duration. \geqslant 30 \sqcap \forall has Duration. \leqslant 60).$$

Another feature of expressive description logics which is missing in $\mathcal{DPCTL}^*$ is qualified number restrictions.

• All learning units start with exactly one introduction which states at least 2 objectives, contains one or more exercises each having between 2 and 5 tasks to solve, and ends with exactly one conclusion:

$$\begin{aligned}&\mathsf{AG}(Beginning Of Learning Unit \rightarrow (\exists^{=1} has Part.\\ &(Introduction \sqcap \exists^{\geqslant 2} presents. Objective))\wedge\\ &\mathsf{E}\,((\exists has Part.(Exercise \sqcap \exists^{\geqslant 2} presents. Task \sqcap\\ &\exists^{\leqslant 5} presents.\ Task))\mathsf{B}\ End Of Learning Unit)\wedge\\ &\mathsf{AF}(\exists^{=1} has Part. Conclusion \wedge End Of Learning Unit)).\end{aligned}$$

On the other hand, due to its abstract semantics $\mathcal{DPCTL}^*$ is more generally applicable and the expressive power of $\mathcal{DPCTL}^*$ exceeds $\mathsf{CTL}_{\mathcal{DL}}$ in some aspects. The following examples taken from [34] cannot be expressed in $\mathsf{CTL}_{\mathcal{DL}}$.

Example 25 ($\mathcal{DPCTL}^*$ Expressions). Mortals are living beings which eventually die:

$$Mortal = Living\_being \sqcap (Living\_being\ \mathcal{U}\ \Box_F \neg Living\_being).$$

Living beings never die out completely:

$$\Box_F \Diamond_F \neg(Living\_being \sqsubseteq \bot)$$

An ordered object must be delivered the next day:

$$Ordered\_object \sqsubseteq \mathsf{A} \bigcirc Delivered\_object.$$

$\mathsf{CTL}_{\mathcal{DL}}$ does not contain any syntactical constructs to express subsumption ($\sqsubseteq$) or equivalence ($=$) relationships between different concepts as done in the formulae above. However, this is not as a serious limitation as it might seem. Recall that $\mathsf{CTL}_{\mathcal{DL}}$ is interpreted on $\mathcal{DL}$

knowledge bases. Hence subsumption and equivalence relationships between document related concepts such as content types or topics can be expressed in the T-Box axioms of the knowledge bases $\mathsf{CTL}_{\mathcal{DL}}$ formulae are interpreted on. The following example illustrates how subsumption relationships can be accounted for within our framework.

Example 26 (Representing Subsumption Relationships). Suppose we would like to enforce the following document properties:

- all *introductions* and *illustrations* help the learner to build up some intuition about the topics discussed;
- content aimed at supporting intuition should not be not too formal;
- *theorems* and *proves* are always formal.

To represent these conditions within our framework we can define the following T-Box axioms in the reference ontology RO (Definition 16):

$$\begin{aligned} &\mathit{Introduc} \sqcup \mathit{Illustr} \sqsubseteq \mathit{SupportsIntuition} \\ &\mathit{SupportsIntuition} \sqsubseteq \forall \mathit{hasFormality}.\neg \mathit{High} \\ &\mathit{Theorem} \sqcup \mathit{Proof} \sqsubseteq \exists \mathit{hasFormality}.\mathit{High}. \end{aligned}$$

Suppose the document contains two narrative units: *intro* which is a highly formal introduction and *illu* which proves some assertion by an illustration. We can represent this scenario by following A-Boxes:

$$\begin{aligned} A_{intro} = \{&\mathit{Introduc}(\mathit{intro}), \\ &\exists \mathit{hasFormality}.\mathit{High}(\mathit{intro})\}, \\ A_{illu} = \{&\mathit{Illustr}(\mathit{illu}),\ \mathit{Proof}(\mathit{illu})\}. \end{aligned}$$

Then the metadata about *intro* and *illu* is locally inconsistent w.r.t. the reference ontology RO (Definition 17) because neither $KB_{intro} = RO \cup A_{intro}$ nor $KB_{illu} = RO \cup A_{illu}$ is satisfiable.

As checking the local consistency of the knowledge representation is mandatory for verifying a document based on $\mathsf{CTL}_{\mathcal{DL}}$ the violations above are discovered in a pre-processing step prior to model checking (see Subsection 5.1).

We have to emphasize that in contrast to $\mathcal{DPCTL}^*$ only subsumption relationships between *static* concepts can be represented within our framework. As content and domain models of documents usually take a static perspective, a temporal formalism is not helpful for T-Box axioms in our application. Moreover, we would run into compatibility problems when sharing ontologies.

Another restriction of $\mathsf{CTL}_{\mathcal{DL}}$ as compared to $\mathcal{DPCTL}^*$ lies in the fact that temporal operators must be combined with a path quantifies. As a result, linear temporal logics are contained in $\mathcal{DPCTL}^*$ but not in $\mathsf{CTL}_{\mathcal{DL}}$. This restriction of $\mathsf{CTL}_{\mathcal{DL}}$ significantly speeds up model checking and simplifies the implementation of a verification framework for $\mathsf{CTL}_{\mathcal{DL}}$. However, we might loose some expressive power for documents having a rather linear narrative structure.

We can summarize the relationship of $\mathsf{CTL}_{\mathcal{DL}}$ to standard temporal description logics as follows:

- as opposed to existing temporal description logics ontological background knowledge represented in $\mathcal{DL}$ T- and A-Boxes can be directly adopted when evaluating the truth of a formula;
- in contrast to general temporal description logics $\mathsf{CTL}_{\mathcal{DL}}$ can incorporate any decidable description logic for representing local properties;
- since the semantics of $\mathsf{CTL}_{\mathcal{DL}}$ is specialized on document structures its expressiveness is not comparable with existing temporal description logics. Neither $\mathsf{CTL}_{\mathcal{DL}}$ can substitute standard temporal description logics such as $\mathcal{DPCTL}^*$ nor vice versa.

7 Conclusion

When integrating fragments from different sources into a web document the consistency of the integrated document in terms of content and narrative structure becomes an issue. We proposed $\mathsf{CTL}_{\mathcal{DL}}$ as an expressive means for the specification of constraints on integrated web documents and showed how $\mathsf{CTL}_{\mathcal{DL}}$ formulae can be evaluated on semantical document models w.r.t. background knowledge in ontologies.

The major contributions of our work are:

- the temporal approach to document knowledge representation does not only support an accurate representation of the narrative structure of web documents but also a loose coupling of distributed knowledge bases such that global properties of document can be verified based even on globally inconsistent knowledge representations;
- the modular structure of $\mathsf{CTL}_{\mathcal{DL}}$ allows for a loose coupling of open world reasoning for the non-temporal part and closed world reasoning for the temporal part of the logic. Hence an efficient implementation using standard systems for reasoning in $\mathcal{DL}$ and model checking CTL as subcomponents is possible and a maximum of compatibility with existing semantic web infrastructure is maintained.

The properties described above make the approach particularly suitable for supporting the integration of heterogeneous documents within semantic web.

References

[1] Bernd Krieg-Brückner, Dieter Hutter, Arne Lindow *et al.* MultiMedia instruction in safe and secure systems. Recent Trends in Algebraic Development Techniques, *LNCS 2755,* 2003, pp.82–117.

[2] Bernd Krieg-Brückner, Arne Lindow, Christoph Lüth *et al.* Semantic interrelation of documents via an ontology. In *Proc. the 2nd E-Learning Workshop Computer Science,* Engels G, Seehusen S (eds.), Paderborn, Germany, Springer-Verlag, 2004, pp.271–282.

[3] Ozsoyoglu G, Balkir N H, Ozsoyoglu Z M *et al.* On automated lesson construction from electronic textbooks. *IEEE Trans. Knowledge and Data Engineering,* 2004, 16(3): 130–140.

[4] Peter Brusilovsky. Adaptive and intelligent technologies for web-based education. *KI—Künstliche Intelligenz,* 1999, 13(4): 19–25.

[5] David Stotts P, Richard Furuta, Cyrano Ruiz Cabarrus. Hyperdocuments as automata: Verification of trace-based browsing properties by model checking. *Information Systems,* 1998, 16(1): 1–30.

[6] Emerson E A. Temporal and Modal Logic. Handbook of Theoretical Computer Science: Formal Models and Semantics, J van Leeuwen (ed.), Elsevier, 1990, pp.996–1072.

[7] David Stotts, Jaime Navon. Model checking cobweb protocols for verification of HTML frames behavior. In *Proc. the 11th Int. Conf. WWW,* Hawaii, USA, ACM Press, 2002, pp.182–190.

[8] Thimbleby H, Addison M A. Intelligent adaptive assistance and its automatic generation. *Interacting with Computers,* 1996, 8(1): 51–68.

[9] Thimbleby H, Ladkin P B. From logic to manuals. *Software Engineering Journal,* 1997, 11(6): 347–354.

[10] Lloyd J W. Foundations of Logic Programming. Berlin: Springer Verlag, 1987.

[11] Christian Süß, Burkhard Freitag, Peter Brössler. Metamodeling for web-based teachware management. In *Advances in Conceptual Modeling, ER'99 Workshop on the World-Wide Web and Conceptual Modeling,* Paris, France, LNCS 1727, Springer-Verlag, 1999, pp.360–373.

[12] Lucke U, Tavangarian D, Voigt D. Multidimensional educational multimedia with $\langle ML \rangle^3$. In *Proc. World Conf. E-Learning in Corporate, Government, Healthcare and Higher Education,* Phoenix, Arizona, USA, 2003, pp.101–104.

[13] Michael Kohlhase. OMDoc: Towards an internet standard for the administration, distribution and teaching of mathematical knowledge. In *Proc. Artificial Intelligence and Symbolic Computation,* Springer-Verlag, LNCS 1930, 2000, pp.32–52.

[14] Amann B, Beeria C, Fundulaki I *et al.* Ontology-based integration of XML web resources. In *Proc. the Int. Semantic Web Conference'02,* Sardinia Italy, Springer-Verlag, LNCS 2342, 2002, pp.117–131.

[15] Laks V S Lakshmanan, Fereidoon Sadri. Interoperability on XML Data. In *Proc. the 2nd International Semantic Web Conference* (*ISWC2003*), Sanibel Island, Florida, USA, LNCS 2870, Springer-Verlag, 2003, pp.146–163.

[16] Philipp Cimiano, Siegfried Handschuh, Steffen Staab. Towards the self-annotating web. In *Proc. the 13th WWW Conference,* ACM Press, 2004, pp.462–471.

[17] Stephen Dill, Nadav Eiron, David Gibson *et al.* SemTag and seeker: Bootstrapping the semantic web via automated semantic annotation. In *Proc. the Int. World Wide Web Conference* (*WWW 2003*), Budapest, Hungary, 2003, pp.178–186.

[18] Siegfried Handschuh, Steffen Staab. Annotation for the Semantic Web. IOS Press, 2003.

[19] Saikat Mukherjee, Guizhen Yang, Ramakrishnan I V. Automatic annotation of content-rich HTML documents: Structural and semantic analysis. In *Proc. the 2nd International Semantic Web Conference* (*ISWC2003*), Sanibel Island, Florida, USA, LNCS 2870, Springer-Verlag, 2003, pp.533–549.

[20] Borislav Popov, Atanas Kiryakov, Angel Kirilov *et al.* KIM — Semantic annotation platform. In *Proc. the 2nd Int. Semantic Web Conference* (*ISWC2003*), Sanibel Island, Florida, USA, LNCS 2870, Springer-Verlage, 2003, pp.834–849.

[21] Niles I, Pease A. Towards a standard upper ontology. In *Proc. the 2nd International Conference on Formal Ontology in Information Systems* (*FOIS-2001*), Chris Welty, Barry Smith (eds.), Ogunquit, Maine, 2001, pp.2–9.

[22] Aroyo L, Mizoguchi R. Process-aware authoring of web-based educational systems. In *Proc. the First International Workshop of Semantic Web for Web-based Learning* (*SW-WL03*), Velden, Austria, 2003, pp.212–221.

[23] Inaba A, Mizoguchi R. Learners' roles and predictable educational benefits in collaborative learning — An ontological approach to support design and analysis of CSCL. In *Proc. the 7th International Conference on Intelligent Tutoring Systems* (*ITS2004*), Alagoas, Brazil, 2004, pp.285–294.

[24] Kasai T, Yamaguchi H, Nagano K, Mizoguchi R. Development of a system that provides teachers with useful resources from various viewpoints based on ontology. In *Proc. the 7th World Conference on Educational Multimedia, Hypermedia & Telecommunications* (*ED-MEDIA 2004*), Lugano, Switzerland, 2004, pp.3349–3356.

[25] Franz Baader, Werner Nutt. Basic Description Logics. In [27], Chapter 2, 2003, pp.47–100.

[26] Ian Horrocks, Ulrike Sattler, Stephan Tobies. Reasoning with individuals for the description logic SHIQ. In *Proc. the 17th International Conference on Automated Deduction,* Springer-Verlag, LNCS 1831, 2000, pp.482–496.

[27] Franz Baader, Diego Calvanese, Deborah McGuinness *et al.* (eds.) The Description Logic Handbook—Theory, Implementation and Applications. Cambridge University Press, 2003.

[28] Alessandro Artale, Enrico Franconi. Temporal Description Logics. Handbook of Time and Temporal Reasoning in Artificial Intelligence, L Vila, P van Beek, M Boddy *et al.* (eds.), MIT Press, 2000.

[29] Alessandro Artale, Enrico Franconi. A Survey of temporal extensions of description logics. *Annals of Mathematics and Artificial Intelligence* (*AMAI*), Kluwer Academic Press, 2001, 30(1-4): 171–210.

[30] Hodkinson I, Wolter F, Zakharyaschev M. Decidable and undecidable fragments of first-order branching temporal logics. In *Proc. the 17th Annual IEEE Symposium on Logic in Computer Science* (*LICS2002*), 2002, 393–402.

[31] Francesco M Donini. Complexity of Reasoning. In [27], Chapter 3, 2003, pp.101–141.

[32] Paolo Bouquet, Fausto Giunchiglia, Frank van Harmelen *et al.* C-OWL: Contextualizing ontologies. In *Proc. the 2nd International Semantic Web Conference* (*ISWC2003*), Sanibel Island, Florida, USA, LNCS 2870, Springer-Verlage, 2003, pp.164–179.

[33] Baader F, Küsters R, Wolter F. Extensions to Description Logics. In [27], Chapter 6, 2003, pp.226–268.

[34] Hodkinson I, Wolter F, Zakharyaschev M. Monodic fragments of first-order temporal logics: 2000–2001 A.D.. *Logic for Programming, Artificial Intelligence and Reasoning,* Nieuwenhuis R, Voronkov A (eds.), LNAI 2250, Springer-Verlag, 2001, pp.1–23.

[35] Wolter F, Zakharyaschev M. Temporalizing description logic. Frontier of Combining Systems 2, D Gabbay, M de Rijke (eds.), Studies Press/Wiley, 2000, pp.379–402.

[36] Sean Bechhofer, Frank van Harmelen, Jim Hendler *et al.* OWL Web Ontology Language Reference, W3C Recommendation 10 February 2004, Mike Dean, Guus Schreiber (eds.), http://www.w3.org/TR/owl-ref/, 2004. last visited Jul. 2005.

Franz Weitl is currently working on his Dissertation in computer science as a research assistant at the Chair of Information Management, University of Passau, Germany. He received his M.S. degree in computer science in 2001. From 2001 to 2004 he has been working at the Institute of Software Engineering and Information Systems on a national project about creating and delivering adaptable XML-based learning content. Since 2004 he has been assisting in teaching applied computer science and internet computing. His research interests are in the application of semantic web technology for intelligent web document management. The aim of his Dissertation is to develop methods and tools to ensure the consistency of documents' overall structure and content on the

semantic level when integrating or reusing (parts of) documents from various sources or generating specialized variants of a document for different application contexts.

Burkhard Freitag has been a professor of computer science since 1994 and chair of information management at the University of Passau, Germany, since 2002. He has studied mathematics at the University of Münster, Germany, and received his Ph.D. degree in computer science from the Technische Universität München (TUM), Germany. Prior to his academic career, Prof. Freitag worked several years with IT-companies and research institutions. Prof. Freitag is also the founder and director of the Institute of Information Systems and Software Technology (IFIS), a technology transfer center of the University of Passau conducting research and development in cooperation with industry and business companies. Prof. Freitag's research interests are in the areas of databases and information systems as well as knowledge and information management. Currently, his work concentrates on context-sensitive data, documents, and processes using methods from formal logic and knowledge representation. Among the technologies applied are database techniques, automated reasoning, object-oriented modeling and programming, and XML languages and tools. Prof. Freitag is a consultant of major companies and institutions and has given numerous seminars and workshops about various topics of computer science and information technology.