Author(s): Nurminen, Miika; Heimbürger, Anneli

Title: Representation and retrieval of uncertain temporal information in museum databases

Year: 2012

Version:

Please cite the original version:

Nurminen, M., & Heimbürger, A. (2012). Representation and retrieval of uncertain temporal information in museum databases. In J. Henno, Y. Kiyoki, T. Tokuda, H. Jaakkola, & N. Yoshida (Eds.), Information Modelling and Knowledge Bases XXIII (pp. 90-109). IOS Press. Frontiers in Artificial Intelligence and Applications, 237. https://doi.org/10.3233/978-1-60750-992-9-90

Representation and Retrieval of Uncertain Temporal Information in Museum Databases

Miika NURMINEN [1] and Anneli HEIMBÜRGER
Department of Mathematical Information Technology
University of Jyväskylä, Finland

Abstract An implementation-oriented model to represent uncertain temporal information in databases is proposed. Temporal information is presented to the user as anchored time intervals with optional beginning and end dates. The model accounts for both instants and intervals, and can be applied to uncertain dates by leaving days and months optional, or by using symbolic constraints that present additional time granularities. The model is defined using conventional relational database structures to support ease of deployment and integration to legacy systems with efficient query capabilities. The model is based on experiences with an existing museum database and highlights challenges related to temporal representation of cultural-historical data in practice. The model is compared with temporal representations used in other museum information systems and collections management standards. Possible opportunities to extend the model in the future research include defining a formal algebraic presentation or utilizing an explicit time ontology.

Keywords. Relational databases, Temporal information, Uncertain information, Cultural-historical information, Information retrieval, Museum information systems

Introduction

Museum information systems (MIS) form a diverse class of collection management and cataloging applications. The content in MIS can document a multitude of domains, including cultural heritage, fine arts, archaeology, science, technology, natural history, sites and monuments, and others [8]. The typical functionality in MIS extends to collections, exhibition and borrowings (loans) management, ad hoc retrieval and reporting, image management, web publishing and integration to other systems, compliance to standard vocabularies and ontologies, user interfaces for staff, public and researchers – even personnel administration [7], depending on the museum domain and extent. Museum collections are characterized by heterogeneity not only regarding their collections, but also due to the different services they provide [20].

In this paper, we introduce a temporal model suitable for a MIS and relate it to temporal theories proposed in other domains and existing museum databases. Perhaps surprisingly, despite the fact that temporal data is an important aspect of museum meta-

[1] Corresponding Author.

data, sophisticated temporal models developed in AI [47] and other disciplines have had relatively little impact on current museum database implementations. Even though the inference mechanisms of a typical MIS is are fairly conventional – often determined by the underlying database – the retrieval and reporting requirements call for versatile data handling capabilities not found in typical business applications. For example, because the number of tables and metadata fields on a MIS can be rather large for a cataloging application (say, tens of tables each containing multiple fields and relationships), and researchers may have unanticipated information needs, adaptible reports and ad hoc query -like capabilities resembling business intelligence systems are often needed.

Cultural-historical information has multiple temporal dimensions (e.g. an object in a collection management system may have descriptive metadata, such as manufacturing date or usage period, as well as contextual metadata [31] related to the cataloging process, such as registration date or cataloging date). Some of this temporal information is typically precise, but uncertain or unknown information exists as well. Even though elaborate temporal databases, query languages, logics and other inference mechanisms exist [47], they do not typically work well with uncertain temporal information (see Section 2.2 on temporal databases), or do not focus on the problems relevant to information retrieval – especially if precise and uncertain information were to be used with the same interface (e.g. constraint propagation, see Section 2.1). On the other hand, while a number of techniques to handle imperfect information have been devised both for databases and AI applications [37], their practical applicability to temporal information is cumbersome from the end user's point of view (e.g. what is the "correct" probability for uncertain endpoints in the temporal interval). Approaches based on temporal *granularities* (algebraic characterization of enumerable mappings to the time domain [6]) seem most promising from the MIS point of view, but even that seems too expressive to be easily implemented in a conventional, SQL-based database. Therefore, more focused temporal model to represent uncertain information – easily applicable with a relational database and intuitive to end users for retrieval – is needed.

We focus on *anchored time intervals* (i.e. periods [29], intervals with fixed start and end dates [22]) to represent temporal information. We do not consider unanchored data (i.e. durations) so for the rest of the paper the short term *interval* is used. Uncertain time is presented to the end user such that certain attributes of the data may be left blank (i.e. the temporal primitive is presented at coarser granularity [4]). For example, we might know the year an old photo was taken, but the exact photographing date is unknown. Similarly, the usage period for a piece of furniture might be known by a decade, but exact years are not known. The intervals are used as a base for a data model that allows retrieving temporal data in an efficient way. The data model is based on the development of the collection management systems in Jyväskylä University Museum (JYU Museum), but could be easily implemented in any relational database application. Based on the requirements gathered from the end users, the model (represented as a UML class diagram and supporting documentation) is evaluated with other prominent collection management software used in Finland, as well as existing museum standards. This paper is structured as follows: Sections 1 and 2 review related research on MIS and temporal information, respectively. Section 3 presents our research case at JYU Museum with the solution outlined in Section 4. Section 5 concludes the paper.

1. Museum Information Systems

From a software engineering perspective, museum information systems contain a rich set of metadata that is comparable in complexity to data structures in ERP (Enterprise Resource Planning) and other business systems. However, considerably less research has been conducted in technical and computational side of museum databases compared to more ”traditional” information systems such as various transaction processing and management information systems (not to be confused with MIS acronym used in this paper), content management systems and other digital repositories. These are considerably more easy to productize and market than MIS because of the sheer number of potential users and relatively streamlined, transaction-like functionality. The situation with museums is also more complex compared to other memory organizations with partially similar tasks but essentially more contained and well-known domains: libraries and archives. Baron [5] points out that developing museum information systems is in a way even more demanding compared to business systems: whereas business systems are usually built around the ability to answer discrete sets of questions, users of databases built to record data for the humanities can never authoritatively predetermine the uses to which the information will be put. This often results to an ad hoc query mechanism, not unlike the functionality used in business intelligence (BI) systems or even expert systems (although the underlying logic and inference mechanism might be less specified compared to actual knowledge-based systems). From usability perspective, most MIS users are non-technical [21]. Therefore, succesful adoption of a collection management system must combine expressive queries and complex data model with a simple and intuitive user interface.

To illustrate the complexity of museum databases, museums of different domains (e.g. cultural history, natural sciences, or art) have widely varying requirements regarding the information stored in the database. In addition, different museums use differing conventions in catalog organization, id numbering, and metadata vocabularies – even the museums of the same general domain. As a result, museum databases are often highly customized, difficult to integrate, and consequently, expensive to develop. The museum databases are in constant transformation themselves. Recently, there has been general trend of shifting from *item-centric* cataloging (physical objects with fixed fields as the primary entities to be cataloged) to *event-centric* documentation, concentrating on the events (e.g. manufacturing, ownership, documentation, publication) related to the objects. Events provide semantically meaningful way of describing links between physical things and actions of human beings, and allow a flexible structure for individual records, splitting the documentation into smaller units that can be added as needed [27]. With little exaggeration it can be argued that every museum database is one of a kind when both database schemas and data entry conventions are taken into account. This makes them both interesting and challenging to analyze.

While the best known museums are large with relatively stable financial base, a majority – and the focus group in this paper – of all existing museums are small institutions [23], possibly grown out of a personal collection and run as a voluntary effort. Since small museums have often limited financial resources (e.g. annual operational budget of less than € 100000) [21], commercial, widely used collection management solutions may have too high a startup cost (but not necessary total cost) compared to a database application developed in-house with minimal features. Even though a few open source collection systems (e.g. CollectiveAccess[2] and CollectionSpace[3]) have recently become available, the typical problems of data conversion (especially when switching from a proprietary database), system configuration, deployment, training, and support remain regardless of the system license type.

There are multiple standardization efforts for museum databases. Among the most prominent are SPECTRUM [44] maintained by the Collections Trust, CHIN Data Dictionaries [12], and CIDOC CRM (Conceptual Reference Model) created by International Council of Museums (ICOM) [14]. UK-based SPECTRUM is recognized as an industry standard for documentation practice, describing not only data fields but also recommended practices for common museum processes (e.g. object entry, acquisition, cataloging, conservation, etc.) as well. The CHIN Data Dictionaries consist of detailed metadata field definitions, that can be used as the basis for a data structure in a collections management system. Data Dictionaries are defined for various domains, including humanities, natural sciences, and archaeological sites. SPECTRUM and Data Dictionaries are based on traditional, item-centric view of the museum collections, whereas the CRM is an ontology featuring a complex conceptual hierarchy, focusing on explicit modeling of relationships [18]. Note that despite the standards and reference models, actual systems in use do not necessary conform to them, as it would stress the costly development effort even further [30] or conflict with existing cataloging procedures. As a result, most MIS adopt the standards partially, like the ArchTerra project [46], or Ida [27]). As an example of the slowly shifting processes, consider that even recently printed storage forms have been used for cataloging [20], and may still be used as a backup method – even in museums that have a collection database.

Integration of multiple databases and publishing the data on the web is a crucial challenge for museum databases and problematic to implement despite the existing standards. Even though CIDOC CRM was developed for data interchange [18], not all systems support it directly, particularly legacy systems that concentrate on item-centric documentation. MuseumFinland[4] is another approach for data integration based on semantic web technologies. the idea is to convert the conventional database data to ontological representation such that the ontology vocabulary is used as field values [28]. The Finnish National Digital Library[5] is an abitious integration effort that attempts to integrate museums, libraries and archives under one search interface. Obviously, no single data format can support all these at once, but NDL offers interfaces for multiple existing formats (e.g. Dublin Core, CRM, SPECTRUM) that are used for harvesting data from the original databases. In practice, the integration may require running two separate databases [46] which can be complicated, especially if data updates from either database were allowed, requiring synchronization. 1-directional transformation [16] from internal representation to chosen standard format is the most feasible approach. Regardless of the integration technology used, data cleaning [11] and preparation for integration is a specific challenge. It has been observed that even 5% or more of the information present in manually created databases is erroneous [9]. While it is tedious to detect and fix in the original database, the situation becomes more complicated if the erroneous data has been published online or integrated to multiple databases.

---

[2] http://www.collectiveaccess.org/
[3] http://www.collectionspace.org/
[4] http://www.museosuomi.fi/
[5] http://www.kdk2011.fi/

2. Time-Aware Applications for Uncertain data

There are a multitude of ways to represent and reason about time. Absolute dates and intervals, temporal relations, temporal logics, temporal databases, or time ontologies can be used to highlight a specific aspect of time with different expressiveness, performance, and inference qualities. Dealing with imperfect (i.e. incomplete, imprecise or uncertain [37]) temporal information adds another layer of complexity to temporal processing. Vila [47] and others [2] have concluded that one temporal model to be used in all domains and applications is a sheer impossibility. One should know the implications of the different models to make informed decisions based on the general domain and application requirements.

There has been remarkably little research on information retrieval with incomplete dates with the data model readily usable with typical museum infomation systems. Only recently, the value of temporal information has been seriously taken into account in conventional information retrieval [3]. Since temporal information provides context for document collections, time can be used to organize the search results regardless of the temporal fields used in the query. Google News Timeline[6] is an example of an IR system with innovative utilization of temporal dimension. In this section, we briefly review some typical temporal applications, considering their applicability in MIS domain.

2.1. Temporal Representation and Reasoning

A classical perspective to reason about time is rooted to logic programming and AI. There are numerous models that the can be used as a basis for implementation. Temporal logics [33] can be used as a foundation for temporal reasoning. The world is modeled as *facts* and *events*. The collection of the facts in a given time is defined as *state*. An important problem related to temporal information in AI is reasoning about actions and change (situation calculus, event calculus) [13], but these are not relevant in MIS context. The general theories to handle imperfect information include probability theory, possibility theory (based on fuzzy logic), and Dempster-Shafer evidence theory [41].

Another popular approach to reason about time is to use an algebraic representation for temporal relations (e.g *before*, *meets*, *overlaps*). Points (instants) [48] or intervals (time spans) [1] can be used as a basis for modeling relations. Models combining both points and intervals have been proposed as well, but these may lead to semantic problems like the divided instant problem [47]. A common theme for these models is to use a graph model where nodes denoting time units are linked with edges labeled with the possible temporal relationships. When new information is added to this representation, it may constrain existing information. In particular, if $e1$ happens before $e2$ and $e2$ happens before $e3$, then $e1$ must be before $e3$. These constraints, based on transitivity, can be used to incrementally update the graph [2]. The technique is called constraint propagation and it has been widely used in AI applications. Generalizations to handle uncertain information have been proposed as well, such as semi-intervals [19], or uncertain temporal relations based on probability theory [41], or fuzzy sets [35].

Temporal Information Processing (TIPS) is associated with natural language processing (NLP) technology to process information expressed originally in natural language, turning it into a ”temporally aware” structured form that can then be reasoned

[6]http://newstimeline.googlelabs.com/

with to solve application-related problems [32]. TimeML[7] is a proposed specification for annotation of events and their temporal anchoring in documents. To enable reasoning in semantic web context, RDF[8] (Resource Description Framework) and ontologies are used for knowledge representation. OWL[9] (Web Ontology Language) is the recommended format for encoding general-purpose ontologies for the web, and OWL-TIME is an OWL extension designed to describe the temporal properties of web pages and web services [36]. Linguistic annotation (without specific temporal focus) has been applied to cultural heritage data as well [28]. Ontologies developed for museum and library domains (e.g. CIDOC CRM [14], see Section 4.3) also include a temporal component.

2.2. Temporal Databases

In relational databases using standard SQL (Structured Query Language), temporal information can be represented basically as dates or timestamps with some variation on precision depending on the system. INTERVAL data type was added to SQL-92 standard to specify durations, but its adoption to actual database management systems has been slow [43]. Manually generated calendar tables can be used to partially compensate for the lack of some temporal functions in SQL (e.g. date arithmetic) [10]. Extensions such as TSQL2 [42] and other temporal Entity-Relationship models [24] have been proposed, but they have not been incorporated to standard SQL. Implementing a temporal database with standard SQL is possible [43], but the queries needed are rather complex.

Characteristic of temporal databases is the usage of date-valued columns as primary keys. This allows one to store the state changes in a given time, preserving the history. The central concepts of temporal databases are *valid time* (the interval during which a fact is true), *transaction time* (the interval during which a fact is stored in the database), and *bitemporal data* for tables that preserve both valid and transaction time [43]. *User-defined time* is supported when time-valued domains for attributes are available in the data model. These are then employed for giving temporal semantics only externally, in the application code and by the database designer [29]. In temporal databases it is assumed that all dates are known, whereas both point- and interval-based temporal models and even models that focus on imprecise temporal information are more concerned with relations between events with little information about concrete dates at all. Even though museum databases hold a great deal of temporal information, most of it is uncertain, and in most systems, no complete history about the data entry updates is preserved. Therefore, museum information systems are, in general, not temporal databases – the temporal data is handled as user-defined time, and the main focus for the MIS is to model mostly unchanging historical information rather than time-varying transactions.

Temporal granularity is important concept that eases the handling of uncertain information in temporal databases. A granularity is an algebraic (operational) or logical (descriptive) [15] characterization of years, months, days, and other enumerable mappings to the time domain [6] that can support both uncertain information and temporal reasoning. Granularities allow one to specify temporal information explicitly at different levels of detail. Relatively user-friendly symbolic notation for customized granularities (e.g. business weeks, *spring*, *beginning of century*, or a specific decade) can be defined using

[7]http://timeml.org/site/
[8]http://www.w3.org/RDF/
[9]http://www.w3.org/TR/owl2-overview/

granularities as well. Goralwalla et al. [22] present a highly expressive granularity model that supports both *anchored* (has specific location on the time axis – practically all data in MIS in anchored) and *unanchored* (relative timespans, durations) temporal primitives. Uncertain time can be represented as *indeterminate* instants for anchored primitives, or indeterminate time spans for unanchored primitives. A common problem is converting temporal data expressed in different granularities such that minimal information is lost in conversion. Numerous models have been proposed for the conversion; a simple approach [4] is to truncate anchored operands in mixed granularities to the coarser granularity to avoid indeterminacy.

From the application designer's perspective, the problem with granularities is partially the same as with other elaborate temporal representations: the more expressive the model becomes, the more difficult it is to represent it with standard SQL constructs. However, the approach seems to be most promising to be used with museum databases since the algebraical way to specify temporal constraints seems to be intuitive to end users. The data model presented in Section 3.2 and revised in Section 4 can be seen as a limited way to present user-defined symbolic time granularities for anchored temporal primitives supporting indeterminacy and conversions between granularities.

3. Collection Management Systems in JYU Museum

In this section, we review the collection management systems used in the Jyväskylä University Museum, focusing on representation and retrieval of temporal information.

3.1. Collection Databases

JYU Museum uses the following applications related to collection management:

- **Duo** includes metadata about photographs, recordings, books and other objects contained in a relatively complex, highly normalized database of ca. 50 tables with 100 fields. Duo has been in production use since 2003, and includes ca. 37000 collection objects (as of January 2011).
- **Arte** includes information about works of art. It has basic functionality similar to Duo, but it is somewhat less complex with ca. 40 tables. Arte has been in use since 2006, and includes ca. 1000 items.
- **Ida** is a next-generation system utilizing event-centric documentation, multiperspective data views, and used-definable schema. Ida is under development. [27]
- Other third-party applications related to collection management (e.g. image processing, web publishing) are also used.

Duo and Arte are implemented as traditional two-tier, database-based client/server applications. They use separate databases, but share most of the code in reusable components (database management, GUI components, search engine) and even parts of the database schema (e.g. contact information and most of the other personal information, temporal data, exhibitions, and image management) [34]. For the rest of this paper, we focus on the Duo database, but the presented temporal model applies to Arte as well.

Duo has the typical functionalities for a museum information system, including three-level grouping for collections (donations, collection records, and collection objects), exhibition and borrowings management, object placement information, keywords, and a multitude of descriptive metadata depending on collection object type (physical object, photograph, recording, book, or newspaper article). Linking to other collection objects, external files, or web URLs is also supported. Personal data about staff, producers, donors, and other people related to object documentation is stored as well. Duo supports ad hoc querying and reporting of most of the fields and tables in the database, including a ”reference search” of any data that points to a specific record (i.e. backlinks). Reports can be generated in HTML format. Duo has also a rudimentary image management facility: preview thumbnails and links to images are stored directly in the database, the original files are stored in a network drive. Automatic import of images is also supported. Support for indirect web publishing via DSpace[10]-based Jyväskylä University Digital Archive (JYX) is currently under development. Integration with the future Ida system is planned.

What sets Duo apart from other museum databases is the high level of normalization in the database schema to keep the vocabulary as controlled as possible using lookup lists extensively – hence the high number of tables. All even vaguely related concepts are collected together to ease searchability. For example, *generic names* field for all collection objects is stored in a separate table instead of a string field directly in CollectionObject (see Figure 1) table – even though generic names vary greatly, especially between object types (e.g. names for physical objects or books are based on the domain, whereas or names for photos and recordings are based on media types). Still, it is useful to provide the user a list of the other values used in current context to reduce spelling and classification errors. Another uncommon aspect of the schema is the decision to use a dedicated table to handle temporal information, resulting to somewhat idiosyncratic retrieval model described in the following section.

3.2. Representing and Querying Temporal Information in DUO

Figure 1 provides a view of the Duo schema focused on tables and fields containing temporal information expressed as a UML class diagram. 1-sided association (open arrows) is used to express one-to-many relationship. For example, CollectionObject table contains a referential key *manufacturingDate* that references a record in DateInterval table containing the actual date. Date attributes (registrationDate, dateCreated, etc.) are standard SQL DATE data types. The most essential tables in the diagram are as follows:

- **CollectionRecord** is used to organize a collection of objects of the same type (e.g. photographs) that belong to the same donation or other acquisition.
- **CollectionObject** is an abstract supertype for all the objects stored in the museum collection with common metadata fields. For example, every object has a manufacturing date, and potentially many producers (e.g. author, photographer, or manufacturer, depending on the object type). The subclasses for each object types add their distinctive metadata fields and other relations.
- **UpdateInfo** is used throughout the essential database tables to track the creation and latest modification dates along with staff members responsible for the changes. The complete modification history or details about changes are not preserved, i.e. Duo does not support transaction time in the sense of temporal databases.

[10]http://www.dspace.org/

- **DateInterval** represents an uncertain temporal interval. For most interval-based metadata fields in the database (e.g. usage date), user can see only years and interval marks in the user interface. For more specific fields (e.g. lifespan), days and months can be edited as well. Any field can be left empty, denoting an unknown day, month, or year. NULL values are deliberately not used because the SQL treatment of NULLs would complicate the queries. Instead, as Celko [10] suggests, an additional value – zero – is used to represent the unknown value internally.
- **PunctuationMark** introduces a number of informal conventions that can not be easily be reflected in searches (i.e. ”-”:normal interval, ”ca.” for uncertain year, and additional symbols for other characterizations such as decades or seasons). If the mark is left empty, only the beginning date should count (i.e. using DateInterval to represent exact date). However, the punctuation mark is currently only descriptive, and does not affect queries.

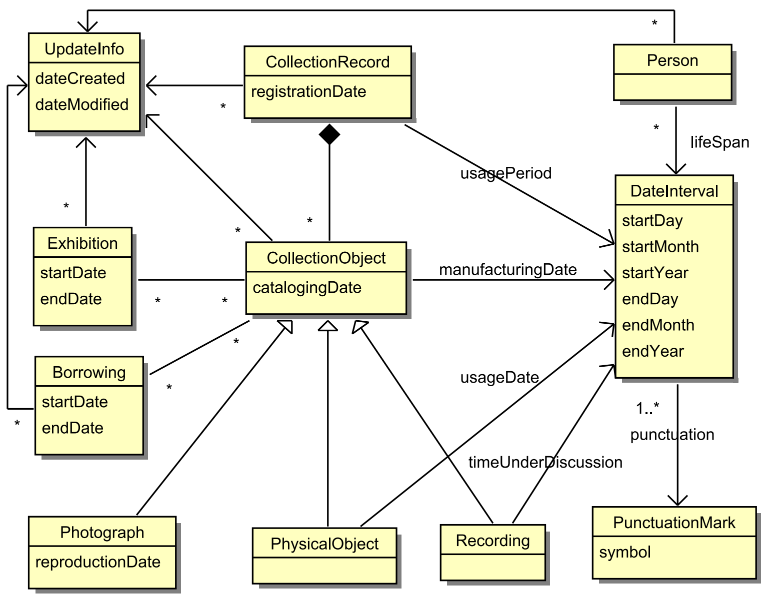

Figure 1. Part of DUO database schema expressed in UML. Intervals and exact dates are decipted as separate entities.

While some of the data related to collection items is relatively certain (especially organization-related events, such as cataloging and modification dates, exhibition dates, and check out dates), allowing to use the standard DATE data type, this is not an option for most of the metadata fields. Much of the historical information is uncertain in the first place, and in many cases the researcher is not interested in exact dates, but the more general temporal periods (e.g. *50s*, *beginning of the century*, *post-war era*) related to the objects. To support this uncertainty with searchable structure, DateInterval and PunctuationMark tables are used to store most of the collection metadata-related information. The user interface for entering dates is depicted in Figure 2.

Fields with DATE data types can be queried in Duo similarly to simple numerical fields: by exact match, or by a given upper/lower bound. For intervals with potentially im-

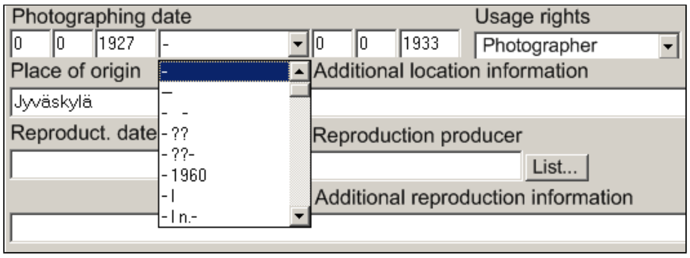

Figure 2. Screen Shot from Duo UI (texts translated from Finnish): entering interval data with all fields visible.

precise information, three search options – each with more relaxed matching constraint – are provided:

- Contained interval (di): matches if both ends of the interval are contained within the query. The most restrictive search option.
- Overlapping interval (od): in addition to contained records, matches intervals that either overlap (or meet) with the query, or contain the query.
- Contemporary interval (ct): in addition to both options above, includes intervals with potentially matching 0-years, such that it is *possible* that part of the result interval operlaps with the query. This way, many search results are matched, among them all records that contain the completely unknown 0-0 interval.

The terminology and abbreviations are adopted from Allen (*during-inverse*, *overlapping/during*) [1] and Freksa (*contemporary*) [19]. Despite the resemblance the relations are not quite identical, because the original models do not consider handling unknown values, which is permitted in Duo both for queries and results. In the result list, start date, interval mark, and end date are compressed to one field. Unspecified years are interpreted as zeros but hidden from the user.

As a detailed example of different interval search options (see Table 1), consider the query $[1990, 2000]$, that matches only i_1 if contained interval search is used. For overlapping interval search, intervals i_2 to i_4 are matched as well, because they overlap or meet the query interval. i_5 does not explicitly overlap the interval because of the unknown start date, and thus does not match the overlapping search. However, i_5 and i_6 are both matched with contemporary interval search, because with unknown values there is a possibility that they are matched.

Table 1. Timespan search example. ct, od, and di indicate different types of matches (contemporary, overlapping, contained) for specific rows with the query specified in the first row.

	Interval		Match
query	`      [---------]`	**1990-2000**	
i_1	`             [--]`	1995-1998	*ct od di*
i_2	`             [------]`	1995-2002	*ct od*
i_3	`             [--------`	1995-0	*ct od*
i_4	`   [-]`	1988-1990	*ct od*
i_5	`----------------]`	0-2002	*ct*
i_6	`------------------`	0-0	*ct*

Note that the timespan query may contain unknown values as well (query $[0, 0]$ matches all records regardless of the match type). For example, if we choose to search all intervals after 1990, the query would be written as $[1990, 0]$ and matches $i_{1..3}$ as contained intervals, $i_{1..5}$ as overlapping, and finally, $i_{1..6}$ as contemporary intervals.

3.3. Problems in Current Approach

While the current approach supports the basic requirements for querying imprecise temporal data, it has several problems listed in this section. The problems are related to the data model, implementation details (i.e. user interface), and the data stored as punctuation marks.

1. First, there are problems with the data model itself. Both the SQL queries and user interface components to represent precise (DATE) and imprecise (DateInterval) temporal data are quite different and cannot be switched in a dynamic way (i.e. a field defined as a precise DATE field cannot be incorporated to imprecise queries). Although the DateInterval table is a general-purpose way to represent imprecise time, the connections from time fields in collection data to DateInterval table are constrained by a rigid, item-centric database schema. New connections can be added to the schema (with code changes required in the client application as well), but the general model is not easy to adapt to event-centric approach: every object is "hard-wired" to manufacturing date and usage date, but new dates or alternate entries for a specific fields cannot be added dynamically.
2. There are also problems related to the user interface. No standard convention is enforced by the system to present single instants in time when DateInterval table is used. By convention, a start date with no interval mark should be interpreted as a point. However, this has not been used consistently and is not directly supported in the query interface (the "point" would be interpreted as a semi-interval with a fixed start date). Besides, the current UI components to represent and search imprecise data are not considered to be intuitive by end users.
3. Finally, the punctuation marks added by staff members are not controlled systematically, apart from showing the list of existing marks. While this allows flexibility in temporal definitions, the lack of control has resulted to many forms of the same interval mark, rarely used marks, and even some erroneous marks. Data quality problems like these are typical in all cultural heritage databases [9], not just with temporal data. After a few manual clean-ups with backtracking references, the number of dubious interval marks has decreased, but as long as the semantics is not precisely defined, applied in the cataloging guidelines, *and* enforced in the user interface, problems will occur. As an example, here is a sample of the interval marks used before the last clean-up ca. 2008.

```
- -- - - - ?? - ??- - 1960 - d - ca.- - c. / /april /spring ? -? -> 0
1937 -1998 2001 2002 7 beg -beginning -beg august before february april
-after december summer -d ? spring beginning -d -d ca ? -d -d.beg d.end
-d? -end - decade -decade- -decade? decades -from decade - from decade?
-end of decade november ca- until ca? circa -circa circa-?
```

Bosch et al. [9] classified typical errors in cultural heritage data to *content errors*, which are genuinely incorrect values; *spelling errors*, which are orthographically corrupted forms of otherwise correct values; and *wrong-column errors*, which are database values that provide correct information but do not fit the database column in which they occur. All of these are present within the loosely controlled punctuation mark.

- Some values such as *beginning*, *-end*, *until* are completely redundant and can be dismissed as content errors (or insufficent guidelines to end users).
- Numbered values (e.g. 1960) are wrong-column errors and can easily be verified by manual checking: the user has inadvertently entered the beginning or end year of the interval to punctuation field. Curiously, the Duo UI would warn the user that a new record would be added to punctuation marks, but values have been added anyway.
- Another common class of punctuation marks are ortographic variations of the "standard" marks defined in original guidelines: - for common intervals, *ca.* for imprecise intervals and *?* for unknown or uncertain values. Most of these are spelling errors, but some combinations of the basic symbols (*-?*, *ca-*, and combinations such as *decade?*) indicate that the (hypothetical) semantics underlining the punctuation marks need to be extended. / is proposed by the museum staff to indicate alternate dates (e.g. interpretation of the interval 1990/2000 as either 1990 or 2000), which is not possible with current search engine.
- The most interesting class of "punctuation marks" are those that indicate a symbolic representation of a specific custom time granularity: *decade*, *summer*, *april*, *d.end* (end of decade). As with other values, different and erroneous spellings are present, but the marks indicate a need to include symbolic, somewhat "fuzzy" information to intervals. However, with the current implementation they are of little use to searching since the punctuation marks are ignored in the interval search.

4. Improved Temporal Model

In this section, an improved temporal model for museum databases is outlined. The model is based on feedback and discussions with end users, review of other museum information systems, and analysis of the data model in Duo. Proposed model should not only bring added value to end users, but also be straightforward to implement without extensive changes to existing database schema or other required components.

4.1. Requirements

The requirements for the improved temporal model can be summarized as follows:

- Performance issues should be addressed. Temporal information is used in almost all end user -specified queries and reports, slowing the application down with complex joins. In addition, the returned datasets can be large.
- The temporal representation should be generalized such that both precise and imprecise information including points and intervals can be marked up and retrieved in a uniform way.
- The number of available punctuation marks should be minimized to keep the model understandable and easy to apply.
- Symbolic information in punctuation marks should affect the interval search and, in some cases, used in place of numerical instants.
- Basic punctuation marks related to uncertainty should be allowed to be combined with other symbolic information.
- Named temporal periods should be allowed to be used as search terms.

- The data model should be brought closer to event-centric documentation to ease integration with CIDOC CRM and other existing applications and ontologies.
- For maintainability, the implemented changes should affect the existing database and code as little as possible to enable smooth transition between versions.
- third-party component usage should be minimized to keep the application as self-contained and easy to deploy as possible.

4.2. Model Definition

Figure 3 presents the proposed schema for the new temporal model as a UML class diagram (cf. Figure 1). The features of the new model are as follows:

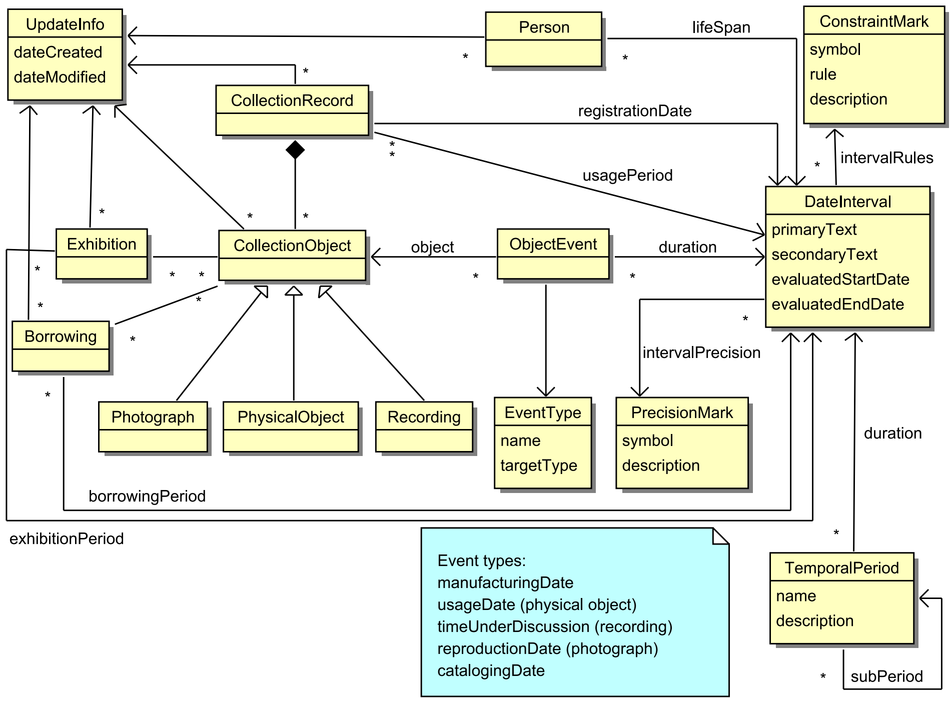

Figure 3. Proposed new schema for temporal entities. ObjectEvent table enables customizable lifecycle-based descriptions for collection objects.

- DateInterval table has been revised such that the values entered by the user (primaryText, secondaryText) are separated from the DATE values that are used in query evaluation.
- PunctuationMark table has been divided into two separate tables to represent symbolic constraints. **PrecisionMark** contains the standard symbols to present uncertainty and imprecision (*?*, *ca.*, optionally with distinct values for each end of the interval), and **ConstraintMark** provides other symbols consisting of custom granularities and other supplemental descriptions as an alternative to be explicitly marked in DateInterval values. The field *rule* is of special importance as it states how the constraint mark affects the actual evaluated field values.

- New tables **ObjectEvent** and **EventType** have been added to support event-centric documentation. Individual DATE or DateInterval-based fields have been omitted from tables with collection metadata and consolidated to ObjectEvent table. This allows adding new events or event types dynamically without changes to schema. *TargetType* field in EventType provides guidelines to the user interface to present a given event type. For example, manufacturing date is applicable to all collection objects, but usage date is by default shown only with physical objects.
- Most of the DATE fields in the rest of the tables have been converted to DateInterval presentation as well to allow uniform search from the temporal fields. The original fields in UpdateInfo are preserved, because the field data is of technical nature, related to database usage rather than collection documentation.
- **TemporalPeriod** table can be used to specify named time periods (e.g. historical eras) in a hierarchical structure. Periods could be used as ”search terms” to quicky retrieve a period of interest during data entry or retrieval, but it is generally not desirable to use them to mark up explicitly a given event. If this was the case, it would be possible to inadvertently change the dates for multiple objects if the period definition changes, which would be error-prone. In addition, the period definitions are not entirely stable (i.e. there is no consensus on precise start and end years of middle ages or renaissance), and the conceptions of time and appropriate historical periods differ in different cultures [26].

The actual evaluation of the user inputs (and possible application of the constraint rules) is left to the application since it would be very problematic to implement them using pure SQL in a portable way. The rule language could be based on regular expressions and simplified date arithmetic, but its detailed definition is left to later development stage. In principle, most of the punctuation marks used in the old model can be used and enhanced with rules. For example, the constraint *decade* would pick all but last number from the beginning year entered by the user, and expand it to evaluated dates such that the whole decade is covered: the user input *1962 decade* would be evaluated to interval [1960-01-01,1969-12-31]. Also unknown days or months affect the evaluated dates: since explicit DATE data type is used in DateInterval table, it would unnecessarily complicate the queries if 0 was still used to represent unknown. Instead, minimum and maximum dates in the given context (e.g. 1000-01-01 and 9999-12-01 in MySql for completely unknown dates) are used. For example, user input *2011-02* would be evaluated to interval [2011-02-01,2011-02-28].

Alternate dates (OR relationship) are not modeled as a separate constraint mark, but by modifying the schema such that an indeterminate number of named object events can be attached to a given collection object. If multiple object events to the same collection object share the same event type, they are regarded as alternatives. This is in line with conventions used in CIDOC CRM (”violation of cardinality constraints is interpreted as an aggregation of alternatives” [18]), and semantic web applications in general.

Although not documented in this paper in detail, a similar approach to event types has already been used in Duo database with manufacturer roles (i.e. a person manufacturing an ”item” can be photographer, artist, director, writer, etc.). With different roles it is possible to model all persons related to the production process, and with the new ObjectEvent table it is more feasible to explicitly document the persons present in a given event in addition to just mark up the date when the event occurred. This also brings the database conceptually closer to CRM.

4.3. Evaluation

From the general schema design perspective the database is still highly normalized. Because of the new tables related to time intervals, and the new ObjectEvent table, it is expected that the query performance will be somewhat slower than before. However, this can be alleviated by creating new indexes and limiting the size of the visible retrieval results. The decision to store both DATE attributes and uncertain periods in a separate table with idiosyncratic rules and external code used in evaluation can be regarded as a disadvantage from the portability standpoint. However, the old model was actually even less portable regarding uncertain intervals, even though it contained fewer tables: the old DateInterval contained the date information with separate INTEGER-valued fields for each date component, whereas in the new model, the *evaluated* interval dates are stored as standard DATE fields. In addition, the evaluated dates already contain the information expressed with constraint marks or uncertain date components. It is yet uncertain whether a regular expression -based language is sufficient, or is a more expressive rule language needed. In the latter case, it would complicate the implementation but would not affect the database schema or evaluated dates.

The most critical limitation of the data model is the lack of semantics within the precision marks. Ideally, a ranked fuzzy or probabilistic search should be applied such that precise dates would get higher ranking than imprecise (ca.) or probable intervals (?). Close matches should be allowed as well (for example, a contained query with endpoint in 1950 might match an imprecise interval ending in 1951). Models for relations between fuzzy time intervals have been proposed [35], but there are multiple practical issues: first, ranked searches can not be naturally implemented within a relational database so a separate search engine would be needed. Second, it is not clear what would be the appropriate ”fuzziness” (level of uncertainty) and shape (deviation for close matches – for example, The Canadian Council of Archives [40] specifies that uncertain intervals should be used only for dates fewer than 20 years apart) of the fuzzy set used to describe the interval. At least it should depend on the original granularities – if both endpoints are described using days, the deviation should be smaller compared to endpoints specified with years only. In addition to the complex implementation, the specification of the intervals would be demanding to the end user. The easiest workaround would be to just set the original temporal query parameters more loosely than originally intended to retrieve more matches and assess the results manually. Precision marks still contain descriptive information for the researcher, even though they are not directly used in the search. Different search options (ci, od, ct) provide a functionality comparable to ranking with varying number of query results depending on the match type.

The new data model implements most of the requirements specified in common library and museum standards regarding the presentation of temporal information, although some of the mechanisms are used or named differently.

- *Dublin Core*[11], a widely used specification of general-purpose core metadata fields is not directly relevant to museum metadata. The fifteen fields specified by the Dublin Core Metadata Element Set are too limited to be used as a primary format for museum documentation – as a publication format, fields in MIS can be mapped to DC. However, the model for representing periods in DCMI Period

[11] http://dublincore.org/specifications/

Encoding Scheme [17] includes a specification for encoding temporal periods in a CSV-like string. The model allows attaching names to periods, and specifying an encoding scheme for the start and end instants. This allows different granularities, calendars and even symbolic names to be used for specifying intervals, and would be a useful extension to the Duo data model.

- *SPECTRUM* [44] contains versatile definitions about temporal intervals. Dates and intervals can be represented with partial information similarly to Duo, and alternate dates are directly supported. A field for textual expression of the date (*Date text*) is also defined. For simple expressions, such as the examples used in the specification (*Late 19th century* or *early 20th century*), constraint marks can be used to similar effect with the added benefit of making the expressions searchable. However, the notion of *qualifiers* to explicitly mark up the probable deviation for start and end instants (e.g. uncertain start date, ± 10 years) is not supported in the Duo data model to keep marking up dates simple. Informally, the notion of uncertainty can be expressed with PrecisionMarks; new marks could be created to explicitly denote the uncertainty in either end of the interval.
- *CIDOC CRM* [14] contains a conceptual hierarchy related to different kinds of events (e.g. birth, creation, transformation) that can partially (but not always for events that are not directly related to collection objects – the most notable exception are persons) be presented with EventType table in Duo. CRM class E4 Period resembles TemporalPeriod table, with period hierarchy interpreted specifically such that containing period *consists of* subperiods. Class E52 Time-Span is analogical to DateInterval in Duo (with E61 Time Primitive as the actual, parseable value for the date ranges), with the addition of qualifiers as in SPECTRUM, and allowing additional temporal relations to be defined between the intervals and other classes (e.g. P81 ongoing throughout, P83 had at least duration, P86 falls within). Without question, the temporal model defined in CIDOC is highly expressive, but very involved to implement or mark up the data if adopted in its entirety. Doerr et al. [18] states that normalization of knowledge to physical reality, such as absolute time does not work, because empirical information is discrete and incomplete, in whatever domain. While the latter is true, we disagree with the conclusion since any numeric search, or integration with applications that do not share the same terminology necessitates a standardized representation format. Duo attempts to solve the problem by preserving both the original user input (with symbolic data in constraint marks), *and* the normalized representation that can be used for retrieval and calculations in application- and ontology-neutral way.
- The conventions for uncertain dates as defined in CCA's Rules for Archival Description [40] are semantically close to the Duo data model with some differences in notation. The authors consider it a good compromise between expressivity and easy markup. *ca.* and *?* denote approximation and probability respectively, dates and intervals are supported, specific years or decades can be left blank with -, OR relation is allowed, and marks can be combined, e.g. `17-?` denotes probable century. In Duo, a similar effect can be accomplished using constraint marks.

Finally, we compare the temporal aspects of the Duo data model to other popular museum software used in Finland. *Musketti*[12], developed by the Finnish National Board

[12] http://www.nba.fi/fi/muskettieesittely

of Antiquities, contains no less than four separate mechanisms for marking up intervals: explicit date interval, explicit year interval, custom time (centuries, decades and other approximate dates), and temporal periods (e.g. middle ages). Even if the exact date interval is known, the equivalent data must be marked to year interval as well [45]. The literal fields for periods and custom time are stored as strings and thus cannot be used for temporal searching without additional parsing – apparently, the numeric dates must be marked up manually. *Polydoc*[13] is another collection management system developed by Redorom InfoSystems. Polydoc contains a simple interval-based mechanism for marking up temporal data, along with optional textual information in a separate field [38]. Web-based, free collection management systems CollectiveAccess and CollectionSpace both claim to support both SPECTRUM and custom schemas, but it is unclear whether a customized rule-like mechanism could be applied for temporal intevals.

The proposed data model is clearly more expressive compared to both Musketti and Polydoc. In addition, constraint marks and a single mechanism for temporal information provides productivity and usability benefits compared to traditional systems. However, the open source alternatives need more thorough investigation regarding both expressivity and customizability.

4.4. Alternative approaches

In this section, we review alternative approaches to enhance the temporal model used in Duo database, reflecting on the techniques and models presented in Section 2. Algebraic representations described in Section 2.1 are relevant in providing standard terminology for temporal relations, but the constraint propagation technique itself is generally not used in museum databases. For each collection object, it is assumed that a numerical – albeit possibly uncertain – dates or intervals are known for documentation. It is clear that museums would benefit from explicit temporal relations between object events. For example, regardless of the actual dates known, object usage date should always be earlier than registration date (when the object is registered to museum collection). However, for other, implicit relations that should be marked up manually, there is a practical productivity problem: when marking up the temporal relations, the number of possible relations is quadratic with the number of events, marking them explicitly takes time, and there are already tens of metadata fields related to a given object. In a case of text annotation with TimeML, Pustejovsky et al. [39] found that due to fatigue humans annotate only a small number of possible links.

Treating the MIS as an temporal database (see Section 2.2) is an attractive prospect. As Snodgrass [43] points out, many applications initially have no temporal component. The need for retaining the history then arises. The first step is to make one or more of the tables temporal, generally by adding an interval timestamp. For example, in the data model used in Duo, tables handling exhibitions and borrowings already contain an interval timestamp to denote the time a given object is in specific state. It is then possible to query the exhibition history for a given object, and this could be interpreted as valid time information. Similarly, a timestamp presenting the latest change in a given record is often used since this allows querying for latest changes in the database. If the database contains a significant amount of errors, it might be desirable to track all changes entered by the user. When the user adds corrections to the database while retaining the original

[13]http://www.redorom.com/polydoc/

data in separate fields, there is always the chance of going back to the original information. The audit trail enables explicit documentation about data cleaning operations (new errors may occur even during the cleaning) [11]. Event-based documentation can also be regarded as an application of transaction time: for example, if a lifecycle (created, sold, damaged, restored, etc.) of a painting is considered [27], these processes are modelled as individual events that affect the state of the painting.

Utilization of general-purpose ontologies (OWL Time, ontologies of MuseumFinland, CIDOC CRM, etc.) for primary data representation would require integrating new software components to the application (e.g. RDF database frontend, inference engine, transformation of existing data). This is a highly sophisticated approach and would be ideal for research purposes, enabling powerful faceted semantic search [28] and expressive inferences on temporal data. However, the approach is laborious to implement, depending on the extent of the ontologies utilized. Utilization of CRM would require mapping of all database tables and part of the individual values (e.g. event types) to CRM classes, and partly the definition of a new ontology to accommodate the specific tables and fields not directly represented in CRM. If more specific ontologies like the ones used in MuseumFinland were adopted, the process is even more complex than CRM-based approach since all *values* used in the original museum database would be mapped to an ontology. Some automation tools exist, but the prosess is not straightforward. Value-level ontologies require considerably more maintenance compared to simpler ontologies based on concepts and relationships alone [25]. As for time ontologies, much of the expressivity provided by them might not be needed in this particular application since the basic interval search with absolute dates has been considered sufficient (although sometimes unintuitive) by the end users. Technically, RDF databases and query languages are not yet mature and stable compared to SQL and relational database technology. In general, we regard ontologies as an important and useful tool for data integration (with primary data still in conventional database and utilizing a 1-way transformation [16]), but even considering the benefits, the development and maintainance cost for a new knowledge base with custom ontological representation is just not realistic.

5. Conclusion

This paper reviewed existing temporal models from the museum information systems perspective. Collection managemenent systems in JYU Museum were introduced and problems with representing and retrieving temporal information were identified. A new temporal model accounting different representations, uncertainty, and customizable events for collection objects was roughly sketched and evaluated with alternative approaches. The model is feasible for implementation, facilitating data entry and uncertain query handling.

Even though it was decided that the representation should be physically in relational database, the system could still utilize ideas from time ontologies (e.g. new query operators). Another important issue that was not possible to be taken into account in this paper involves the implementation of the user interface – even though the data model was created based on discussions with the museum staff, further feedback on the new model and usability testing is still required. As usual, the user interface must be developed in cooperation with the end users. What kind of visual component would be the most intuitive and flexible to be used for temporal queries and data entry?

Future research involves the implementation of the model in a relational database. Current data with old temporal model in the production database must be transformed to the new representation. Careful planning concerning the current punctuation marks is required – while some obvious errors can be corrected with simple replacement operations, a few ambiguous marks still remain that should be corrected before updating the database to keep it as uncluttered as possible from the start. The software should assist the user with the punctuation marks so that new, unnecessary marks are not added. Mapping rules and temporal periods for culture-sensitive (uncertain) temporal information and generalizing the model to different calendars would also be an interesting prospect.

Acknowledgements

The authors thank the Finnish Academy and the Japan Society for Promoting Science for supporting our research. Acknowledgements to Lasse Saari, Ari Häyrinen and other staff in Jyväskylä University Museum for ideas and discussion about the proposed model.

References

[1] J.F. Allen: *Maintaining knowledge about temporal intervals*. Commun. ACM, **26**(11):832–843, 1983.

[2] J.F. Allen: *Time and time again: The many ways to represent time*. Journal of Intelligent Systems, **6**(4):341–355, 1991.

[3] O. Alonso, M. Gertz, and R. Baeza-Yates: *On the value of temporal information in information retrieval*. SIGIR Forum, **41**:35–41, 2007.

[4] F. Barbic and B. Pernici: *Time modeling in office information systems*. SIGMOD Rec., **14**:51–62, 1985.

[5] R.A. Baron: *Choosing museum collection management software the systems analysis: Its methods, functions and benefits*, 1991.

[6] C. Bettini *et al.*: *A glossary of time granularity concepts*. In O. Etzion, S. Jajodia, and S. Sripada (eds.): *Temporal Databases: Research and Practice*, 406–413. Springer, 1998.

[7] J. Blackaby and B.E. Sandore: *Building integrated museum information retrieval systems: Practical approaches to data organization and access*. Archives and Museum Informatics, **11**(2):117–146, 1997.

[8] P.L. Boeuf *et al.*: *Integrating museum & bibliographic information: The Sculpteur project*. In *ICHIM 05 – Digital Culture & Heritage*, 2005.

[9] A. van den Bosch, M. van Erp, and C. Sporleder: *Making a clean sweep of cultural heritage*. IEEE Intelligent Systems, **24**(2):54–63, 2009.

[10] J. Celko: *SQL for Smarties: Advanced SQL Programming , Third Edition*. Morgan Kaufmann, 2005.

[11] A. Chapman: *Principles and methods of data cleaning – primary species and species-occurrence data, version 1.0*. Techn. rep., Global Biodiversity Information Facility, Copenhagen, 2005.

[12] *CHIN data dictionary - Humanities*. Techn. rep., Canadian Heritage Information Network, 2010. `http://www.pro.rcip-chin.gc.ca/bd-dl/ddrcip_sh-chindd_hs-eng.jsp`.

[13] L. Chittaro and A. Montanari: *Temporal representation and reasoning in artificial intelligence: Issues and approaches*. Annals of Mathematics and Artificial Intelligence, **28**:47–106, 2000.

[14] *Definition of the CIDOC Conceptual Reference Model, version 5.0.2*. Techn. rep., International Council of Museums, 2010. `http://www.cidoc-crm.org/official_release_cidoc.html`.

[15] C. Combi, M. Franceschet, and A. Peron: *A logical approach to represent and reason about calendars*. In *Proceedings of the Ninth International Symposium on Temporal Representation and Reasoning (TIME'02)*. IEEE Computer Society, 2002.

[16] O. Cure and J.D. Bensaid: *Integration of relational databases into owl knowledge bases: demonstration of the DBOM system*. In *22nd International Conference on Data Engineering (ICDE) – Workshops*, 230–233. IEEE Computer Society, 2008.

[17] *DCMI period encoding scheme*. Techn. rep., Dublin Core Metadata Initiative, 2006. `http://dublincore.org/documents/2006/04/10/dcmi-period/`.

[18] M. Doerr, C.E. Ore, and S. Stead: *The CIDOC conceptual reference model – a new standard for knowledge sharing*. In J.C. Grundy *et al.* (eds.): *Tutorials, posters, panels and industrial contributions at the 26th Int. Conf. on Conceptual Modeling (ER 2007)*, 51–56. Australian Comp. Soc., 2007.

[19] C. Freksa: *Temporal reasoning based on semi-intervals*. Artif. Intell., **54**(1):199–227, 1992.

[20] D. Gavrilis, G. Tsakonas, and C. Papatheodorou: *Designing interoperable museum information systems*. In *14th International on Virtual Systems and MultiMedia (VSMM)*, 2008.

[21] G. Geser and A. Mulrenin: *Are small heritage institutions ready for e-culture?* In *ICHIM 04 - Digital Culture & Heritage*, 2004.

[22] I. Goralwalla *et al.*: *Temporal granularity: Completing the puzzle*. Journal of Intelligent Information Systems, **16**:41–63, 2001.

[23] P. Gray: *Who are you calling cheap?* In *Museums and the Web 2007*. Archives and Museum Inf., 2007.

[24] H. Gregersen and C.S. Jensen: *Temporal entity-relationship models - a survey*. IEEE Trans. Knowl. Data Eng., **11**(3):464–497, 1999.

[25] A. Halevy, P. Norvig, and F. Pereira: *The unreasonable effectiveness of data*. IEEE Intelligent Systems, **24**:8–12, 2009.

[26] A. Heimbürger: *Temporal entities in the context of cross-cultural meetings and negotiations*. In *Information Modelling and Knowledge Bases XX*, 290–308. IOS Press, 2009.

[27] A. Häyrinen: *Towards semantic modelling of cultural historical data*. In *Information Modelling and Knowledge Bases XXII*, 312–320. IOS Press, 2011.

[28] E. Hyvönen *et al.*: *A content creation process for the semantic web*. In *Proc. of OntoLex 2004*, 2004.

[29] C. Jensen *et al.*: *The consensus glossary of temporal database concepts (February 1998)*. In O. Etzion, S. Jajodia, and S. Sripada (eds.): *Temporal Databases: Research and Practice*, 367–405. Springer, 1998.

[30] B. Kelly, M. Guy, and A. Dunning: *Addressing the limitations of open standards*. In J. Trant and D. Bearman (eds.): *Museums and the Web 2007*. Archives and Museum Informatics, 2007.

[31] V. Lyytikäinen: *Contextual and Structural Metadata in Enterprise Document Management*. PhD thesis, University of Jyväskylä, 2004.

[32] I. Mani, J. Pustejovsky, and B. Sundheim: *Introduction to the special issue on temporal information processing*. ACM Transactions on Asian Language Information Processing (TALIP), **3**:1–10, 2004.

[33] D. McDermott: *A temporal logic for reasoning about processes and plans*. Cognitive Science: A Multidisciplinary Journal, **6**:101 – 155, 1982.

[34] M. Nurminen *et al.*: *Duo & Arte documentation*. Techn. rep., University of Jyväskylä, 2004. `http://users.jyu.fi/~minurmin/duo/`, in Finnish, retrieved 2011-01-28.

[35] H.J. Ohlbach: *Relations between fuzzy time intervals*. Proceedings of the 11th International Symposium on Temporal Representation and Reasoning (TIME'04), 44–51, 2004.

[36] F. Pan: *A temporal aggregates ontology in OWL for the semantic web*. In *AAAI Fall Symposium on Agents and the Semantic Web*, 30–37. AAAI Press, 2005.

[37] S. Parsons: *Current approaches to handling imperfect information in data and knowledge bases*. IEEE Trans. on Knowl. and Data Eng., **8**:353–372, 1996.

[38] *Polydoc web documentation*. Techn. rep., Redorom InfoSystems, 2006. `http://www.redorom.com/polydoc/info/webpages.pdf`.

[39] J. Pustejovsky *et al.*: *ARDA summer workshop on graphical annotation toolkit for TimeML*. Techn. rep., NRRC, 2003. `http://nrrc.mitre.org/NRRC/TangoFinalReport.pdf`.

[40] *Rules for archival description (RAD),*. Techn. rep., Canadian Council of Archives, 2008. `http://www.cdncouncilarchives.ca/public.html`.

[41] V. Ryabov: *Handling Imperfect Temporal Relations*. PhD thesis, University of Jyväskylä, 2002.

[42] R.T. Snodgrass (ed.): *The TSQL2 Temporal Query Language*. Kluwer, 1995.

[43] R.T. Snodgrass: *Developing Time-Oriented Database Applications in SQL*. Morgan Kaufmann, 1999.

[44] *SPECTRUM: The UK museum documentation standard, version 3.2*. Techn. rep., Collections Trust, 2009. `http://www.collectionstrust.org.uk/spectrum`.

[45] S. Valanto: *Musketti documentation*. Techn. rep., The National Board of Antiquities, 2003. `http://www.nba.fi/fi/musketti_luettelointiohjeet`, in Finnish.

[46] V. Vassilev *et al.*: *Museum information systems: CIDOC data model implementation in the ArchTerra project*. Bollettino Del Cilea, **69**, 1999.

[47] L. Vila: *A survey on temporal reasoning in artificial intelligence*. AI Commun., **7**:4–28, 1994.

[48] M.B. Vilain and H.A. Kautz: *Constraint propagation algorithms for temporal reasoning*. In *Proceedings of the Fifth National Conference on Artificial Intelligence (AAAI-86)*, 377–382. AAAI Press, 1986.