OpenSDE: A strategy for expressive and flexible structured data entry

Renske K. Los*, Astrid M. van Ginneken, Johan van der Lei

Department of Medical Informatics, Ee2157, Erasmus MC, University Medical Center Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands

Received 18 April 2004; received in revised form 8 April 2005; accepted 8 April 2005

1. Introduction

Electronic patient data are associated with many potential benefits, e.g. data sharing, decision support, quality assessment, research, and management of patient care [1–4]. The degree to which patient data are currently available electronically varies. To harvest the potential benefits of electronic data, the data must also be structured to enable processing by computer applications [2]. Structuring the medical narrative poses a significant challenge: content and level of detail are often unpredictable and vary per domain (and even per clinician) [5]. In an attempt to structure medical narratives in a manner that allows for variation and unpredictability, we have developed OpenSDE (SDE: structured data entry). OpenSDE is an application that supports clinicians with the recording of structured data for use in both care and research [6,7]; data that are till now typically recorded in free text narratives.

Other published work on support of SDE does not provide much insight in the functionality and expressiveness of the respective applications. Therefore, in our description of OpenSDE we focus on those aspects that enable flexibility and expressiveness in data recording. Since OpenSDE is based on the selection of predefined concepts, we also explain why we did not choose to directly adhere to an existing terminology standard. OpenSDE is available in open source [6].

Summary

Purpose: This description focuses on the expressiveness and flexibility of OpenSDE: an application that supports recording of structured narrative data.

Methods: OpenSDE enables data entry with (customizable) forms based on trees of medical concepts. The relevant scope for data entry can be tailored per medical domain by construction of a domain-specific tree. OpenSDE is intended for structuring narrative data to make these available for both care and research.

Results: The OpenSDE application is currently in use at several departments in our academic hospital, including radiology, neurology, pediatrics, and child psychiatry. OpenSDE is available for all in open source.

© 2005 Elsevier Ireland Ltd. All rights reserved.
2. Medical narratives

The medical narrative can be found in diverse sections in the medical record: medical history, family medical history, physical examination, progress notes, and reports (e.g., radiology, surgery, or pathology reports) [8]. Medical narrative data tend to be unruly [5], and only predictable to a certain degree. Free text has been the ideal format to collect these data as it has a high degree of expressiveness [9]. Free text allows clinicians to record data in whatever words, abbreviations, or codes desired. The challenge is to structure the medical narrative in a manner that poses no a priori limitations on detail and that structures the data in a manner also suitable for research. An additional aspect that must be addressed for data sharing purposes (and multi-disciplinary research, for example) is uniformity in data representation. An application that attempts to tackle the challenge must be generic, yet tailor able to specific domains, to allow data sharing and domain specific data collection.

3. OpenSDE: goal and perspective

The goal of OpenSDE is to support structured data entry in a variety of settings, so as to have patient data available for both routine care and retrospective and prospective research. This implies that OpenSDE intends to support two goals that have diverging requirements for data format and level of detail. For care, data are preferably entered as free text, whilst, for research purposes, one prefers coded data.

When developing an application like OpenSDE, one must choose a perspective from which to approach the problem [9]. The direct benefit of structured data lies primarily with the research component. However, one is dependent on the clinicians for data collection, and to motivate them to structure data, benefits, such as validity checks or report generation, must be added. Data collection for research as a separate activity from data collection for patient care would be an undesirable expansion of the clinician’s task. We, therefore, developed OpenSDE from the perspective of care aiming to provide seamless integration of data collection for research.

4. OpenSDE: data entry

The expressiveness provided by the OpenSDE application must not pose (a priori) limits on the level of detail in which one wants to structure data. Not only should data entry be highly expressive, it should also be straightforward. OpenSDE, therefore, applies the following principle for structured data entry. Data can be entered about predefined concepts. These concepts are organized as nodes in a tree structure (we refer to this as a domain model). In this tree, every node is described by its sub-tree, as shown in Fig. 1. In general, the deeper one navigates into the tree the more detailed a sign or symptom can be described; the tree also holds constraints relevant for the presentation of data entry options. The essence of data entry with OpenSDE is traversing the tree of medical concepts and selecting those nodes that correspond with the medical observations. The tree is domain specific; the modeling of trees and tree characteristics is discussed in the paragraphs about domain models, further on in this paper.

4.1. Entry forms

Fig. 1 shows a screen capture of the OpenSDE data entry application. The left-hand side shows the tree that contains the predefined medical concepts. In this example, ‘history of present illness’ is the selected concept. The right-hand side of the screen illustrates a (standard) form for the concept ‘history of present illness’. This form contains all concepts in the sub-tree of ‘history of present illness’, i.e., all concepts that are relevant to describe in the context of the history of the current illness. If, however, the sub-tree of a concept is more than three levels deep, the form becomes too large to oversee. Therefore, after the third level in depth, we use hyperlinks to subdivide a form into more detailed forms. At the bottom of the form in Fig. 1, a hyperlink is presented for the concept ‘patient uses anticoagulants’.

The example in Fig. 1 shows entered data for history of present illness in its corresponding form and in the tree. As one navigates through the tree on the left, the forms will change accordingly. The form always corresponds with the selected concept in the tree, and is generated by the application, based on the concepts in the tree. Making changes to the tree does not require manual adaptation of the standard forms.

Users can create custom forms that contain the medical concepts relevant to a particular scenario.

---

1. OpenSDE has its roots in ORCA (Open Record for Care) [10]. Since 1996 the structured data entry module has been separated from ORCA as a stand-alone application. This SDE-application underwent many changes in the subsequent years. Since March 2003 the SDE-application is available in open source as OpenSDE [6].
OpenSDE: A strategy for expressive and flexible structured data entry

In a custom form for a particular concept, the user can select a sub set of the nodes in the sub-tree of that concept, and determine the order in which the selected nodes appear on the custom form. A custom-made form for, e.g., a diabetes check-up may be defined to contain such concepts as blood pressure, weight gain and loss, eyesight, sensibility, and other relevant information. Clinicians can define the forms to accommodate specific topics and their individual preferences, which enhances the flexibility for data entry. Custom forms can be made using a form editor, which is a tool inside the OpenSDE application.

4.2. Expressiveness in data entry

We will use the example provided in Fig. 1 to illustrate the kind of expressiveness that forms the basis for OpenSDE.

A clinician admitting to the ER a patient with a trauma to the head caused by a blunt object, will need to record data relevant to the scenario. Relevant data may include the mechanism and time of injury; any accompanying symptoms such as headache or vomiting; loss of consciousness recalled by patient, bystander, or companion; and findings from the physical examination and radiologic investigation.

In the example above, mechanism and time of injury are descriptors of history of present illness in the sense that they describe the injury in more detail. The accompanying symptoms, such as headache and vomiting can be present or absent. Furthermore, time of injury requires the recording of a date/time value, whereas the duration of headache requires a numeric value with a unit.

In OpenSDE we support these examples of expressiveness in a generic way. Besides the ordering of medical concepts as nodes in a tree, each
node has a set of data items to specify, for example, presence state (absent, present, or unknown), time-stamp, and value. The presence state is entered in the check box in front of the medical concept, as can be seen on the form in Fig. 1. A check is present, a cross is absent, a question mark means unknown, and an empty check box implies that no data have been recorded about the concept.

An additional data item to enhance expressiveness is the main complaint, which enables the clinician to label vomiting as ‘main complaint’ as this was the reason for encounter. It may also be necessary to describe the progression of a complaint over time, e.g., headaches have become less frequent. Besides that, a complaint or symptom may manifest itself differently in different circumstances. Headache, for example, may be local in the morning and diffuse in the afternoon. In other situations it may be necessary to record distinct data about the left and the right ear: bleeding is present in the right ear, but absent in the left. To structure such data we have enabled the user to duplicate particular sub-trees to allow the recording of the chronology, different manifestations, and multiple instances, respectively. The actual storage of the data sub-trees and all other data recorded with OpenSDE is described elsewhere [11].

When clinicians feel the need to record data that cannot be represented by any of the data items offered in OpenSDE, they can add comments in free text.

4.4. Domain models

The OpenSDE application is generic in the sense that it can be tailored to multiple specialisms without the need for changes to the database and software. The data entry procedure with OpenSDE is the same for every user, regardless of the specialism. The only difference is that the content varies per specialism.

OpenSDE uses domain models, which are trees of medical concepts purposely developed for the application. A specific domain model is created per medical discipline. A cardiology domain model, for example, contains all the relevant concepts at the necessary level of detail for the cardiologist to record his medical narrative data. In general, these models do not contain knowledge needed for inference, such as ‘fracture affects bone’. ‘a skull is a bone’, therefore, a skull can have a fracture. The aim of domain models is to define the concepts and constraints that are relevant to record the medical narrative.

Domain models vary in content from each other but not in terms of structure, i.e., the model for cardiology will vary in content from the model for pediatrics but the representation (structure) of the content remains the same. The domain model should, therefore, be seen as consisting of a content and a structure. The content refers to the medical concepts that can be selected during data entry to create medically relevant expressions, whereas the structure refers to the tree format in which these concepts are represented. Domain models are manually authored.

4.5. Domain model structure: trees

The domain models are represented as a rooted tree structure. A rooted tree consists of nodes and arcs that connect these nodes, and has one root node. Every node, except the root, has one parent node, while every parent node may have one or more child nodes. A node without children is called a leaf. For every node, one path extends from the root to the particular node.

4.6. Domain model content: medical concepts

The developers intended the OpenSDE application to be used for the recording of medical narrative data. The content of the domain model for patient contacts, although tailored per specialism, will generally contain the sections: patient history, family history, review of systems, and physical examination
Every section contains elements that are more specific: the deeper one navigates into the tree, the greater the level of detail. The hierarchical character of domain models reflects the nature of medical descriptions.

In Fig. 2, the concept 'Neurologic findings' is described by concepts as 'lateralising motor weakness', 'lateralising sensory disturbances', and 'focal neurological deficit'. Lateralising motor weakness, can be either 'left' or 'right'. The path that leads from the root to a node indicates the context in which that node should be interpreted. In the case of Fig. 2, 'left' belongs to 'lateralising motor weakness', but 'left' can also be used in the context of 'lateralising sensory disturbances'.

4.7. Domain models: data constraints

Creating domain models (modeling is described in Section 4.8) not only encompasses defining and ordering the concepts about which data can be recorded, it also includes defining the constraints on the data. Constraints include the type of information that can be entered about a concept (presence states, numerical or free text values, etc). Constraints also include the limits and restrictions on the data (plausible values for systolic blood pressure must be in between 90 and 200, and the systolic pressure must be higher than the diastolic pressure).

Constraints are added to nodes by giving the node a 'node type' and by assigning appropriate...
properties to each node. A node can be one of four types: feature, option, unit, or shortcut. The properties that can be set depend on the node type of a node.

The node type ‘feature’ represents a characteristic that cannot be entered as absent; blood pressure and weight are features since a person always has a blood pressure and a weight. The presence state of a feature is, therefore, either present or unknown, and never absent. In the case of features, OpenSDE only accepts presence state ‘present’ in combination with a further description.

A concept receives the node type ‘option’ when it is an optional item of data: something that can be entered either as present or absent (or unknown), such as ‘headache’. The main difference between a feature and an option is that the presence state of a feature cannot be ‘absent’ whereas for an option it can be.

Fig. 2 presents an extract of the domain model used for the radiology/neurology study. The concept ‘post traumatic amnesia’ (PTA) can be described by ‘duration’, which has been modeled as child of PTA. The node PTA has the node type ‘option’, as it is something that is not necessarily present in a patient. The node ‘duration’ has been modeled as ‘feature’ node type. This characteristic is always applicable when a person has suffered from PTA; if there is PTA, it always has a duration.

The ‘unit’ node type is always a child node of a node with a numeric value property, and indicates the possible unit(s) of the value that must be entered. Fig. 2 illustrates that the units of ‘duration’ in this example, are seconds, minutes, or hours. A unit node can be further specified by ‘default unit’ and ‘unit factor’. In Fig. 2, the default unit is set to ‘minutes’. The unit factor enables calculations between the default unit and the other possible units.

Sometimes a finding is relevant in more than one medical context, i.e., should be offered for description in more than one place in the tree. Edema of the extremities, for example, may be relevant in the context of cardiovascular, renal, or endocrine disorders. It is, however, not desirable to describe the same finding in more than one branch in the tree. Instead of describing the same edema in more

<table>
<thead>
<tr>
<th>Property</th>
<th>Effects on data entry</th>
<th>Feature</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core entity</td>
<td>Represents the main entity of interest in one particular path</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Description mandatory</td>
<td>The presence state of at least one of the concept’s children must be entered, or a comment must be added</td>
<td>+ By default</td>
<td>+</td>
</tr>
<tr>
<td>Description requires evidence</td>
<td>If concept is described further, at least one of its children must be present, or a comment must be added</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Multiple instance</td>
<td>Applies when multiple occurrences of a concept can be described (e.g., fingers, warts)</td>
<td>NA</td>
<td>+</td>
</tr>
<tr>
<td>Multiple description</td>
<td>Allows more than one description to be added to describe the concept in different circumstances</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Value</td>
<td>A value is any one of the following:</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Numeric</td>
<td>Concept is a numeric value of the type: single numeric value; value that lies within a range in the form of ( x \pm y ) or value has a margin ( x \times y )</td>
<td>+</td>
<td>NA</td>
</tr>
<tr>
<td>Calculated field</td>
<td>A numeric value may contain a calculation based on values of other nodes. This enables the calculation of scores (e.g., APGAR or GCS)</td>
<td>+</td>
<td>NA</td>
</tr>
<tr>
<td>Free text</td>
<td>Allows entry of free text data</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Moment</td>
<td>Allows date or date-time value</td>
<td>+</td>
<td>NA</td>
</tr>
<tr>
<td>One child present only</td>
<td>Only one of the child nodes may be ‘present’ (for mutually exclusive children)</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Condition</td>
<td>Data must conform to a specific condition (systolic pressure &gt; diastolic pressure)</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Picture</td>
<td>A picture can be added to illustrate a specific concept</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Codes</td>
<td>Concept to which a classification code, for example an ICD-10 code, can be added</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

*Signifies that the property is applicable (i.e., it can be set by the modeler); + ‘By default’ means the property is set by default; NA stands for property not applicable.
than one place, only one of the edema nodes contains all relevant describing child concepts, and the other edema nodes become references to this node. We refer to this reference as a ‘shortcut’. The node type ‘shortcut’ is conceptually different from the node types ‘feature’ and ‘option’. The feature and option node types represent certain constraints on the data that can be recorded about a node. Shortcuts were added for the convenience of data entry, and to prevent the same concept from being described twice in a structured manner in a tree.

The node type determines which properties can be assigned to the nodes. The two most frequently occurring node types in the domain models are ‘feature’ and ‘option’. The properties that can be set for features and options are listed in Table 1, together with a brief explanation of the implications of these properties for data entry.

There are two properties that require more explanation: core entity and codes. In every path, one node is assigned the core entity property. This identifies it as the main node of interest in this path. As mentioned in the paragraph on expressiveness in data entry, we enable the user to duplicate particular sub-trees to allow the recording of the chronology, different manifestations, and multiple occurrences of an observation. This duplication of sub-trees is only allowed at the level of core entities and, if more than one type of duplication applies, in a predefined order. This limitation to the expressiveness was introduced to increase predictability. If these sub-trees are allowed anywhere in the tree and can be nested in any order, the way in which data will be recorded becomes highly unpredictable, which for our purposes is undesirable.

The codes property allows a code to be assigned to a particular node. This enables a link to a classification or terminological system. If desired, codes can be shown in the OpenSDE interface (see Fig. 1). Codes are described in more detail in the discussion of this paper.

4.8. Creating domain models

Domain models are created by experts in particular fields of medicine, using a specifically designed tool (Domain Model Editor), which creates a visual representation of the concepts as they are ordered in the tree structure. Modeling is described in more detail elsewhere by Doupi and van Ginneken [12]. When creating a new domain model, the domain model editor will display only one node: the top (or root) node. The domain model is expanded by first adding a child node to the top node. New nodes are then added to this node as siblings, meaning that they are on the same level (as ‘history of present illness’ and ‘physical examination’ are in Fig. 2), or as children of the node (in Fig. 2, ‘local’ is a child node of ‘headache’).

When adding a new node to the tree, the node must be assigned a node type and the applicable properties, as described above.

5. Discussion

Since OpenSDE domain models are trees of predefined concepts, domain models intuitively resemble a terminology. Therefore, we often receive the question what the difference is between domain models and a terminology, or why we did not use a terminological system instead of our own, manually authored, domain models.

Standardization is essential for the aggregation and pooling of data for clinical research, as well as for the sharing of data between applications that need to process the data [13]. Data structure is important for research and decision support. With the OpenSDE application, we want to enable data collection for research purposes, as well as for the sharing of data between applications that need to process the data [13]. Data structure is important for research and decision support. With the OpenSDE application, we want to enable data collection for research purposes, as well as for the sharing of data between applications that need to process the data [13]. Data structure is important for research and decision support.

1. There is an essential difference between the goal of a terminological system and the goal of OpenSDE domain models. Terminological systems have less granularity than the requirements that patient care poses on OpenSDE domain models, because terminology standards usually support granularity at levels appropriate for aggregation of data.

2. Standards are rigid, which is of course part of their purpose.

The first reason for not committing to a terminological system is the difference in goal between terminological systems and OpenSDE domain models. Terminological systems are mainly intended for semantic matching of medical concepts so as to enable data aggregation, exchange, and reasoning about concepts. For example, the aim of the GALEN Project was to construct a reference terminology
in a formal representation that allows reasoning with general knowledge about what "can be said" [14], as well as semantic matching involved in pooling of data. According to the National Library of Medicine, UMLS is intended to "facilitate the development of computer systems that behave as if they "understand" the meaning of the language of biomedicine and health, ... UMLS is not optimized for particular applications" [15]. If we were to choose a subset of the UMLS we would currently not be able to fully meet the requirements for documentation of narratives. Using such tools, therefore, still require much manual adaptation. OpenSDE is specifically intended to document the patient’s signs and symptoms in detail. As a result, the information contained in OpenSDE domain models differs from the information contained in a terminological system.

Secondly, terminological systems have less granularity than the requirements that patient care poses on OpenSDE domain models, because terminological standards usually support granularity at levels appropriate for aggregation of data. Treatment decisions and care providing in general requires more detail than the terms used in many terminological systems [16]. Besides, few terminologies contain all relevant concepts to describe the medical narratives of all domains.

The last reason for not committing to a terminological system is that standards are rigid, which is of course part of their purpose. If one wants to support data entry for multiple specialisms and accommodate new insights and procedures in a flexible manner, it must be possible to add, change, or remove concepts (no longer) necessary for data collection. Altering a standard can take years, which in a practice setting is undesirable. If a modeler insists on using a terminological system when creating an OpenSDE domain model, three strategies could be followed. The first would be to limit the concepts in the OpenSDE domain models to the relevant terms from a terminological system. As described above, this would be unsatisfying due to the limited level of granularity; data collection is limited to particular concepts, which may not suffice for specific clinical research. A second option would be to choose one terminological system and expand it with those concepts, details, and relations needed for OpenSDE in a care setting. This not only requires adding new terms or a greater level of detail, it may also require expanding the formalism of the terminological system with new relations or concepts types that are needed for supporting structured data entry [17]. However, due to the rigidity of standards, this option is far from ideal. A third option is to create the domain models with concepts to suit those that will use them and, where relevant, associate these concepts with concepts of a terminological system, for example, by using codes. It will not, however, be possible to map all concepts from the domain models to one particular terminological system, unless a terminological system is created that contains all concepts necessary for describing the medical narrative in a particular domain. For OpenSDE we chose the third option as this gives the modeler the freedom to map concepts to relevant concepts from a specific terminological system and to combine these with other concepts to create a domain model suitable for his purpose.

OpenSDE and its relation to standards proposed by CEN and HL7 has been addressed elsewhere [17]. As mentioned, OpenSDE is designed and intended to support structured data entry in a variety of settings, in such a way that no redundant recording is required to have patient data available for both routine care and retro- and prospective research. OpenSDE is not the only development that aims at enabling the above. However, documentation of expressiveness and functionality of similar SDE applications is scarce or outdated, making it difficult to obtain a good overview of state-of-the-art SDE applications. From the available literature, it is difficult to distill whether or how systems such as UltraStar [18] or PenIvory [19] deal with different manifestations of complaints in different circumstances and the extent to which clinicians are free to choose the level of detail in which they structure their data. An opportunity to compare applications like Medcin [20], Pen & Pad [21], IMR-E [22], Pure MD [23], UltraStar [18], PenIvory [19], and Purkinje [24] in terms of expressiveness, functionality, and use of standards may be very useful to the SDE community.

OpenSDE is currently being used by several departments at the Erasmus MC, including pediatrics, immunology, and child psychiatry, and is available in open source [6]. Alongside, five pilot studies in a clinical practice setting are being undertaken to evaluate OpenSDE in terms of: completeness of domain models, uniformity of data representation, and acceptance by end users. Depending on the outcomes of these studies, a decision will made about whether OpenSDE will be made available throughout the academic hospital. The pilot studies are in the domains of venerology, ear, nose and throat, pediatrics, liver disease and transplants, and anesthesiology. The study performed in conjunction with the departments of radiology and neurology started in February 2002 and ended in November 2004. Data have been recorded for over 1800 patients. Both the collected data and the data format were suitable
OpenSDE: A strategy for expressive and flexible structured data entry

for evaluating the decision rule under investigation in the study [25].

6. Conclusion

Approaching structured data entry from the care perspective places emphasis on approaching the expressiveness of free text. We chose this perspective because we wanted to ensure that data collection corresponds as much as possible to the needs of the clinicians who are actually recording the data. Having spent effort on enabling data entry in a manner that suits clinicians, the next step is to approach the challenge from the perspective of research. Is it possible to use data that are unpredictable and potentially diverse? How can the hierarchically organized data be extracted for scientific analysis? Such questions are addressed in the paper entitled: “Extracting Data Recorded with OpenSDE: Possibilities and Limitations” [26].

Summary points

What was already known:
1. Electronic patient data are associated with many potential benefits, e.g. data sharing, decision support, quality assessment, research, and management of patient care.
2. Structuring the medical narrative poses a significant challenge: content and level of detail are often unpredictable and vary per domain (and even per clinician).
3. Standardization is essential for the aggregation and pooling of data for clinical research.

What did this study add to our knowledge:
1. A comparative study of various approaches and tools for SDE has to be performed as to better understand the pros and cons of these approaches.
2. Despite the variety in medical domains, the narratives have many requirements for documentation in common. We found a solution to allow the recording of the chronology, different manifestations, and multiple occurrences of an observation.
3. Current terminologies are not detailed enough to serve as the basis for structured recording of narratives. Although coding systems may be used, additional terms are needed to allow the richness required in clinical care.

Acknowledgements

The authors would like to thank Cobus van Wyk, Georgio Mosi, and Jan Talmon for their helpful comments on previous versions of this manuscript. The work presented in this paper is funded by a grant from ZonWw.

References