# Models for Forms

Daniel Abler
CERN
1211 Geneva 23
Switzerland

Charles Crichton, James Welch,
Jim Davies, Steve Harris
Department of Computer Science
University of Oxford
United Kingdom

## ABSTRACT

To make reliable, safe, and effective use of data outside the context of its collection, we require an adequate understanding of its meaning. In data-intensive science, as in many other applications of computing, this necessitates the association of each item of data with complex, detailed metadata. The most important, most useful piece of metadata is often a description of the form used in data acquisition. This paper discusses, with examples, the requirements for standard metamodels or languages for forms, sufficient for the automatic association of form data with a computable description of its semantics, and also for the automatic generation of form structures and completion workflows. It explains how form models in specific domains can be used to facilitate data sharing, and to improve data quality, and semantic interoperability.

## 1. INTRODUCTION

In many areas of human endeavour, progress is increasingly dependent upon the acquisition and analysis of large amounts of data. In science, medicine, and government, data from many different sources is combined to test theories, to predict outcomes, and to inform decisions. In each case, the adequacy of a test, the accuracy of a prediction, and the soundness of a decision may depend upon the quality of the data involved, the application of an appropriate analysis, and the correct interpretation of results.

There are many aspects of data quality. [22] lists several aspects under three main "dimensions"—intrinsic, contextual, and representational. The aspects that concern us here correspond to each of these dimensions. They are: *correctness*, the extent to which values entered correspond to the intended interpretation; *completeness*, the extent to which the data collected is complete; and *comprehensibility*, the extent to which the data comes with adequate documentation—other data that we may use to determine whether the data is fit for a specific purpose, whether a particular application is appropriate, and whether an interpretation is correct.

The design of *forms* for data acquisition has a bearing upon all three aspects. A well-designed form can make the intended interpretation of each value accessible or even obvious to the user, and hence promote correct data entry; it can also provide a degree of validation prior to submission, by checking that the values entered are of the appropriate type, and that values in different fields bear an appropriate relationship to each other.

Similarly, a well-designed form can promote completeness, by helping the user to navigate between sections, pre-populating fields with default or existing values, and offering an appropriate selection of questions and responses. Finally, a good design can promote comprehension, by ensuring that the resulting data is linked to appropriate *metadata* describing the questions, the responses, and the context of submission.

All of this makes forms an ideal subject for domain-specific modelling and model-driven engineering. They make a significant contribution to data quality: any investment in improving their design through new abstractions and new languages is likely to be rewarded. By factoring out common elements of form design, and providing mechanisms for generating and configuring implementations from abstract models, we can make it easier to produce and deploy well-designed forms.

A model for a form, written in a well-defined language or conforming to a shared metamodel, is a valuable item of data in its own right. Used as an item of linked metadata, it can provide valuable semantic information about any data that the corresponding implementation is used to collect. Furthermore, form models can be used in advance of data collection, as a means of planning and coordinating data collection activity: ensuring that the required questions are asked, and that no question is asked unnecessarily.

In this position paper, we will explore the requirements for a metamodel or language of forms. In Section 2, we consider examples from the domain of clinical research. In Section 3, we consider the different aspects of forms that we would wish to address in a metamodel, and outline our progress to date.

## 2. CASE REPORT FORMS

Medical research, like the practice of medicine, relies upon detailed clinical observations. To determine the value and effectiveness of a particular therapy, we need to record detailed information about treatments and outcomes in a large number of patients. The need for large numbers arises from the range of factors that may affect outcomes, and the complexity of the mechanisms that govern the progress of diseases such as cancer. For each possible factor, we require a sufficient number of patients for whom that factor is present, and a sufficient number for whom it is not, before we can make any reliable, statistic inference about its effect.

Obtaining these numbers is difficult. For any particular disease, we might agree upon a standard set of questions or observations, but if we wish to advance our understanding, develop new treatments, or test new theories, there are always new questions to be asked. There is no single, standard form to complete. Indeed, each study, investigation, or clinical trial may involve many different forms: asking different questions, or asking similar questions under different circumstances.

An investigation based upon a single trial, or a single set of medical records, will often fail to produce statistically significant results. This is particularly true for early-phase studies, or experimental medicine, in which detailed observations are made of small numbers of patients; these may suggest new explanations, new treatments, even cures, but the evidence from a single study is rarely conclusive.


**Figure 1: fragment of a case report form**

It is important, therefore, to be able to combine information from multiple studies: to identify comparable observations, and to integrate or transform the corresponding data values to produce a larger evidence base. The increasing use of electronic forms for data capture should facilitate this: an electronic record of the form used can be stored along with the data. Figure 1 shows a "case report form" from a vaccine study: the contextual information provided by the form—for example, the fact that this is information recorded on the first visit—may be important in understanding and interpreting the values recorded.

When data integration is possible, the results can be dramatic. For example, data from 400 different trials of the drug *Tamoxifen*—many of which appeared to produce contradictory results—allowed researchers to identify the subset of the population responsive to the drug, and indicated the optimum period of treatment. This evidence changed clinical practice in the UK, and reduced mortality from operable breast cancer by 24%.

Most attempts at data integration, however, are less successful. Even if the data itself is available, it is often too difficult or costly to establish whether or not the observations are comparable and—where comparability is established—to implement the appropriate data transformations. Even if electronic forms are used, any need for *manual intervention* in locating and interpreting documentation, or in writing queries and programs for data extraction and transfer, represents a significant barrier to progress.

Public sector and industry organisations have attempted to overcome this barrier in two different ways: through data documentation standards, and through common platforms for data capture. Examples of the former approach include the Data Documentation Initiative (DDI) [23], and the Clinical Data Interchange Standards Consortium (CDISC)'s [2] Operational Data Model (ODM) [10]; examples of the latter include the *OpenClinica* [1] and *REDCap* [4] study management systems.

Neither of these approaches have produced entirely satisfactory results. The data standards activity has been focussed upon *post-hoc* documentation: models of forms are used to record form contents and structure, but not to generate forms for data acquisition. As a result, modelling represents an additional burden upon researchers, who may derive no tangible benefit themselves from the increase in data quality and re-usability. Furthermore, unless the form models are generated automatically from the documentation, or vice versa, it is likely that the description of the data afforded by the model is inaccurate.

Conversely, although study management systems such as *OpenClinica* or *REDCap* use form models as the basis of form generation and deployment, these models are relatively simplistic: researchers provide input in the form of a spreadsheet, with entries in different columns representing question text, answer range, value constraints, and navigation rules. What is more, the models are not presented as documentation for the data captured, and are difficult to re-use outside the context of the particular study management system.

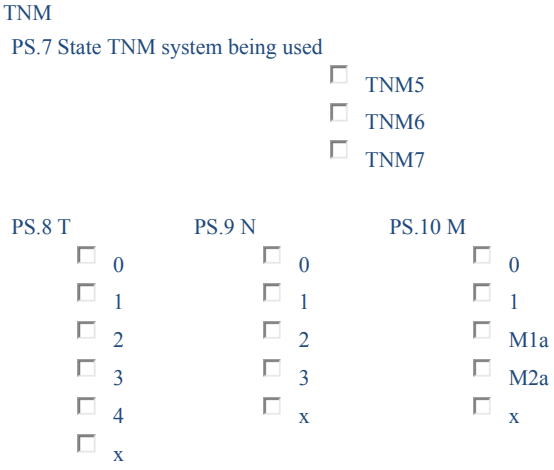


**Figure 2: a fragment of an *OpenClinica* form**

For example, the fragment of a case report form produced using *OpenClinica* shown in Figure 2 above was generated from a spreadsheet, part of which is shown in Figure 3. Other columns of the spreadsheet specify the header, subheader, grouping, and layout information, as well as the expected response types. In this case, there are four enumerated types to introduce; each of these is entered as a comma-separated list of values in the appropriate column of the spreadsheet.

| ITEM_NAME | DESCRIPTION_LABEL | LEFT_ITEM_TEXT | UNITS |
|---|---|---|---|
| A_2a_PS_TNMSystem | Which TNM system is being used | State TNM system being used | |
| A_2a_PS_TumourValue | Tumour value (T in TNM) | T | |
| A_2a_PS_NodeValue | Node value (N in TNM) | N | |
| A_2a_PS_MetastasisValue | Metastasis value (M in TNM) | M | |
| A_2a_PS_EUSTStage | EUS T stage | EUS T stage | |
| A_2a_PS_EUSMStage | EUS M stage | EUS M stage | |

**Figure 3: a fragment of an *OpenClinica* spreadsheet**

Neither of these approaches—neither common documentation standards, nor study platforms—has resulted in a standard means of recording the logical relationships between questions or sets of questions asked in different forms or in different studies, except by reference to a shared data standard, data dictionary, or "question bank". To determine whether two observations are comparable, it is not enough to check that they are based upon the same "standard" question: the additional context provided by the form may be decisive. Conversely, two observations may be comparable even when the question text is quite different: our comparisons should be based upon semantic, rather than syntactic, identity.

## 3. METAMODELLING FORMS

We may consider the elements of form design in three parts, according to the purpose of the constraints involved.

1. *identitification and logical structure*

   (a) In order that we may classify and refer to data captured using a form, we must be able to uniquely identify each data component. For example, confronted with a date value entered into a US customs form, we would wish to know whether this corresponded to the date on which the form was completed, or the date of birth of the person completing it.

   (b) We will need also to identify sets or groups of data components. Different components may be associated for many reasons: it might be that two observations together correspond to a blood pressure measurement, or that a particular set of observations on one form is to be compared with a particular set of observations on another. Such an association need have no bearing upon the presentation of the form.

2. *data constraints*

   (a) The values entered against different data components on the form—the answers to different questions—may be related. For example, if one question asks about the number of children a person has had, and a subsequent one asks about the number of grandchildren, then we might expect an answer of "0" to the first to lead to an answer of "0" for the second. We may include logical constraints in the design of a form to capture such properties, providing further information about the intended interpretation of the data.

   (b) If we have also some notion of submission then these constraints become universal properties of any data set corresponding to a submitted form. More generally, a data constraint (or validation rule) could be associated with any event in a form completion process: the completion of a section, the return of a form for modification, or the value of some metadata item associated with the submitted form (a warning flag, perhaps). To facilitate this, we require some means of identifying the data constraints declared.

3. *process or presentation constraints*

   (a) Our interpretation of the data collected by a form may be influenced by the way in which the form was presented: the order in which the questions were asked; the way in which the answers were provided (by means of a "radio button", for example, or a text box); and any default answer or partial completion offered to the user. The documentation standards and the study management systems that we have considered all include some form of "question flow" or "skip logic", and for good reason.

   (b) Conversely, process or presentation constraints form an important aspect of form design. The usability of a form, and the quality of the resulting, may be greatly enhanced by omitting questions that are rendered irrelevant by earlier answers, by offering only valid responses, or by pre-populating fields with default or already-determined values.

Our purpose in providing a means of identifying each form component is so that we might sensible delegate the responsibility of further documentation to some "semantic context". The role of this context, and the nature of the linked documentation, will depend upon the kind of data involved. In the case of clinical data, there are recommendations provided by the CONSORT [5] group (for clinical trials) and by CDISC [2] (for data interchange and reporting in the pharmaceutical industry). For example [14]:

> *An instrument used to generate, capture, transfer, manipulate or store source data (e.g. Case Report Form (CRF). . . ) shall be an accurate representation of the protocol ensuring that the data as specified within the protocol is captured correctly.*

As a clinical protocol describes not only the form but also the circumstances and consequences of its completion, we should expect to link our electronic forms to documents describing workflows and other aspects of study design. A general metamodel for forms should support arbitrary, additional metadata by allowing the identification of relevant components within a form instance.

The DDI standard [18] does particularly well in this respect. Figure 4 shows core properties inherited by every XML "type" defined within DDI: these correspond to questions, forms, datasets, study descriptions, and value domains—everything described in a DDI document has a unique reference, allowing arbitrary semantic constraints to be described and imposed.

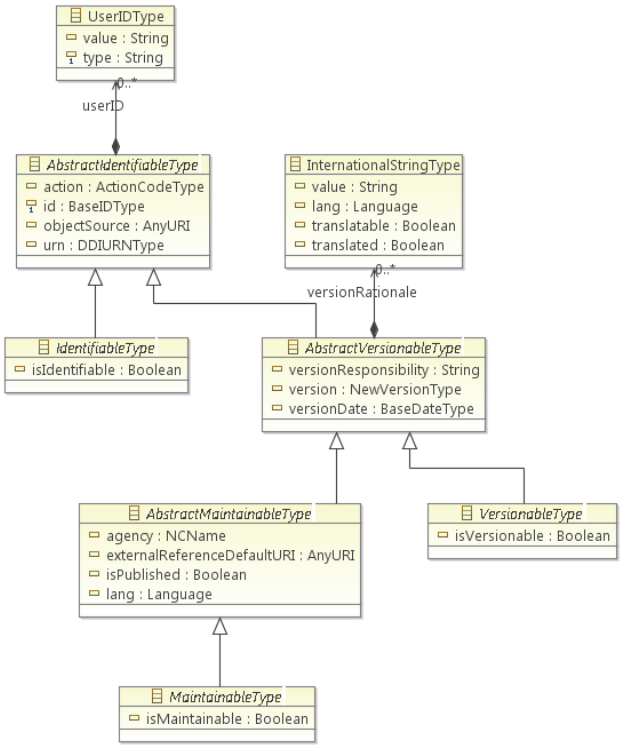


**Figure 4: Identification and Versioning in DDI**

The DDI proposal includes also a schema for describing semantic constraints or relationships. In its current version, this is neither fully abstract (it includes a specific weighting scheme, for example) nor constructive, but it is nevertheless indicative of the kind of complementary metamodel or language that we require for subsequent reasoning about form models and data.

An additional feature of the DDI proposal is the explicit treatment of versioning and maintenance. Whether we need to include this in a metamodel for forms depends upon whether we can assume that the metadata items that we refer to will be maintained indefinitely. If we assume the existence and availability of metadata registries—as implementations, perhaps, of the ISO 11179 standard for metadata registration [17]—then it should not be necessary to "in-line" this metadata.

It is a simple matter to carry forward these references to a form implementation. For example, the case report form shown in Figure 1 was generated using a standard forms application, *Microsoft InfoPath* [3], from an XML schema—itself generated, again using InfoPath, by instantiating a metamodel for clinical trials, described in [12]. The schema, and the generated InfoPath form, both include `SAWSDL` [24] identifiers, as "model references": see Figure 5.

```
<xs:complexType
    name="Visit1Registration"
    sawsdl:modelReference="...">
  <xs:sequence>
    <xs:element
      ref="v1r:informedconsentformsigned"
    ...
    <xs:element
      ref="v1r:participantFitforImmunisation"
       ... />
  </xs:sequence>
</xs:complexType>

<xs:simpleType
  name="informedconsentformsigned"
  sawsdl:modelReference="
    https://cdebrowser.nci.nih.gov/
    CDEBrowser/search?elementDetails=9
    &FirstTimer=0&PageId=ElementDetailsGroup
    &publicId=2004073&version=4.0">
  <xs:element
    name="informedconsentformsigned"

    type="v1r:informedconsentformsigned">
    ...
  </xs:element>
  <xs:restriction base="xs:string">
    <xs:enumeration value="no" />
    <xs:enumeration value="yes" />
  </xs:restriction>
</xs:simpleType>
```

**Figure 5: Embedded SAWSDL identifiers**

The importance of being able to reference each component of the form can be explained in terms of the likely pattern of re-use. Even if the questions on a form have been drawn from a shared data standard or question bank, their meanings will have been extended or modified by placing them in the context of the form. To determine whether two observations in different studies are comparable, we need to consider and refer to the various components of that context. As [7] observes, almost any aspect of form design may have an effect upon the meaning of a question or response.

Some of the form components will have a particular role. The most obvious of these are the *data component*s, corresponding to questions or sets of questions. Each of these should be linked to a specification of possible responses: a *response type*. These two classes (or metaclasses) of component are common to every modern forms language or metamodel; they are also at the heart of the ISO/IEC 11179 standard for metadata registration, in the form of "data elements" and "value domains".

We would propose to adopt a more general, more abstract approach, with a language of data components and response types that is properly *compositional*. This is not true of questions or data elements in, for example, ISO 11179, where each element has properties—such as fixed, explanatory text, or its place in a single domain ontology or object-class-property hierarchy—that cannot sensibly be combined. By considering simply the central aspect of data components—their identification, and their associations with ranges of possible values—we can usefully compose them.

For example, we might sensibly identify a section of a form as a data component, composed of a number of questions, each of which is also identified as a data component. The response type of the form section may be derived—as a type union—from those of the questions involved. A fragment of the proposed metamodel is shown in Figure 6.

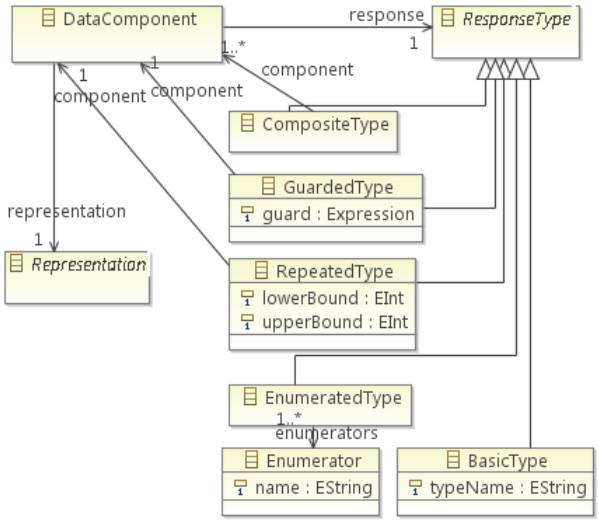


**Figure 6: Data Components and Response Types**

Each data component can be associated with additional constraints upon values. We might do this for several reasons, but the most important is that of describing a guard or validation constraint for successful form submission. Constraint information, like response types, can be composed in a straightforward fashion: submission constraints for individual data components can be combined to produce submission constraints for form sections or complete forms; they can also be re-used within form workflows.

Another kind of composition applies to the process or presentation constraints. The representations of "skip logic" or "question flow" in study management systems such as *OpenClinica* and *REDCap* are not compositional, relying on "go to" constructs and conditions that include references to question numbers, constructions that make sense only at the level of a complete form. We would propose instead to define a language for the sequential and parallel composition of data components as workflows.

Our intention is that this language should allow for the use of form data components as forms in their own right:

- a single form might be broken apart and completed by means of a more complex, perhaps iterated, workflow
- the question flow language within a form could mirror a workflow language in which form submissions were workflow actions

- a set of forms might be factorised to ensure that the same question isn't asked more than once: either by pre-population of form fields, or by modifying structure or process constraints.

# 4. PROGRESS

Practical work on modelling and generating electronic forms dates back to the late 1970s and early 1980s: see, for example, [16] and [15]. These were relatively simple efforts, aimed at the generation of basic user interfaces for databases. After the introduction of the microcomputer during the 1980s many systems for collecting data became available [9]. Modern programming language environments contain DSLs for describing the composition of the user interface and how it links into the program code, a good modern example is XAML [20]. However, the manual programming involved means that the forms produced using these technologies are costly and difficult to maintain.

Languages such as HTML, and applications such as Adobe Acrobat, define their own standards for presenting forms and handling the submission of data. In the case of HTML, form fields can be accessed using scripting languages—typically, Javascript—that provide question flow, validation, and submission functionality. However, the level of abstraction afforded by these approaches is relatively low, with no accessible, declarative description of semantics or intended functionality.

XForms [25] and InfoPath [3] provide a higher level of abstraction. These models use XPath to bind data from forms controls, such as text entry fields, into an explicitly provided instance of XML, which can be used to prepopulate the form with existing data. The presentation of forms in both these systems relies upon existing web-based technologies such as HTML, XSLT, and cascading stylesheets.

An XML Schema of the instance model enables XForms or InfoPath to perform basic validation of data and to provide appropriate form controls for the type; additional validation rules on the data that is bound can be included declaratively in the forms definition. Non-declarative scripts can be triggered by events linked to the forms—such as data entry an form submission.

In the UK CancerGrid [8] project, working in partnership with the US NCI caBIG programme and the US Department of Veteran's Affairs, we decided to adopt InfoPath and XForms as the basis for forms generation and semantic annotation in support of clinical research activity in cancer. For various reasons, we were unable to deploy the technology in cancer during the lifetime of the project: this was due in part to investment, institutional and personal, in other, more labour-intensive means of producing forms.

We were however able to deploy the technology in support of vaccines research in the UK and in Nepal [13], with some success: we were able to reduce the time taken to produce a complete set of semantically-annotated forms from six months to as many weeks: not least because a set of prototype forms could be shown to clinical researchers within a few days, while conversations on their design were still current.

We were able also to use some of our generation technology to annotate spreadsheets, as form models, for cancer clinical studies run on the *OpenClinica* system. While this did not offer the same level of automation, it did ensure that the primitive data components of the deployed forms were properly associated with unique identifiers, each linked to a published metadata item in an ISO-standard registry.

The XML produced in XForms and InfoPath is limited to a fixed-depth, pre-defined form structure. Our attempts to generate more sophisticated forms were doomed to failure: an domain-specific language encoded as an XML schema containing an expression language cannot be implemented using this approach without additional, bespoke scripting, which would defeat the objectives of semantic interoperability and re-use.

For this reason, we are embarking upon the design of a core forms metamodel, in collaboration with colleagues in the ISO/IEC JTC1 SC32 WG2 working group. Our intention is to link the features of the metamodel to related standards for datatypes and semantic metadata registration. As suggested above, we intend to provide for clear separation between logical composition and forms presentation or execution. We intend also to provide a compositional language for internal workflow, and to support related notions of form composition and refactoring in the context of external workflows.

With the aim of providing technology that can be used directly by the researchers, scientists, and domain specialists involved, without the need for technical support from a software engineer, we would hope also to produce or re-use intuitive, graphical notations for the description of semantic and structural relationships, as well as for completion workflow and validation constraints.

The types of systems we are trying to represent contain structural elements that can be described using classes, associations, and constraints, but also behavioural elements such as completion and validation workflows that are better expressed using a programming notation or expression language. We are interested in ways of building and maintaining systems containing both these styles of construct; versioning, change, and evolution are particular concerns, and we expect to adopt ideas from [19] and [21].

A further, practical concern is the referential transparency of semantic references and assertions. We need to make explicit assumptions about the immutable nature of published data objects, and about versioning procedures, in order that we may copy external metadata within our forms for convenience or archiving purposes. The notion of *linked data* proposed by Berners-Lee [6] needs to be refined to deliver the kind of guarantees about data quality and traceability that we require in science and medicine—and will soon come to expect in other application areas.

Along with the core metamodel we would propose to define a number of candidate implementations or specialisations, focussed upon accepted practices and standards in clinical studies management, electronic health records, and open government. An additional area of interest lies in the customisation of our languages and metamodels based upon specific domain models.

# 5. ACKNOWLEDGEMENTS

Daniel Abler and Steve Harris would like to acknowledge the support of the EU FP7 PARTNER (215840) and ULICE (228436) projects, respectively. Charles Crichton would like to acknowledge the support of Microsoft Research. All of the authors are grateful to the anonymous referees for their comments.

# 6. REFERENCES

[1] OpenClinica https://community.openclinica.com/. 2011.
[2] Clinical Data Interchange Standards Consortium - CDISC http://www.cdisc.org/, 2011.
[3] Microsoft InfoPath http://office.microsoft.com/en-us/infopath/, 2011.
[4] Research Electronic Data Capture - REDCap http://project-redcap.org/, 2011.
[5] D. G. Altman, K. F. Schulz, D. Moher, M. Egger, F. Davidoff, D. Elbourne, P. C. Gtzsche, and T. Lang. The

revised CONSORT statement for reporting randomized trials: Explanation and elaboration. *Ann Intern Med*, 134, 2001.

[6] C. Bizer, T. Heath, and T. Berners-Lee. Linked data - the story so far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 2009.

[7] A. Bowling. Mode of questionnaire administration can have serious effects on data quality. *Journal of Public Health*, 27(3):281–291, September 2005.

[8] R. Calinescu, S. Harris, J. Gibbons, J. Davies, I. Toujilov, and S. Nagl. Model-driven architecture for cancer research. In *Software Engineering and Formal Methods*, 2007.

[9] E. Carpenter. Software tools for data collection: microcomputer-assisted interviewing. *Social Science Computer Review*, 6(3):353, 1988.

[10] CDISC. Specification for the Operational Data Model (ODM) Version 1.3.1. Technical report, Clinical Data Interchange Standards Consortium (CDISC), 2010.

[11] H. Cho and J. Gray. A domain-specific modeling language for scientific data composition and interoperability. In *Proceedings of the 48th Annual Southeast Regional Conference*, ACM SE '10, pages 107:1–107:4, New York, NY, USA, 2010. ACM.

[12] J. Davies, J. Gibbons, R. Calinescu, C. Crichton, S. Harris, and A. Tsui. Form follows function: Model-driven engineering for clinical trials. In *International Symposium on Foundations of Health Information Engineering and Systems*, 2011.

[13] J. Davies, J. Gibbons, S. Harris, J. Metz, A. J. Pollard, and M. Snape. Model-driven support for a vaccine study in kathmandu. In *Microsoft eScience Workshop*, 2009.

[14] GCP INSPECTORS WORKING GROUP. Reflection paper on expectations for electronic source documents used in clinical trials. Technical report, European Medicines Agency, London, UK, 2007.

[15] N. Gehani. The potential of forms in office automation. *IEEE Transactions on Communicatons*, 30(1):120–125, 1982.

[16] M. Hammer, W. G. Howe, V. J. Kruskal, and I. Wladawsky. A very high level programming language for data processing applications. *Commun. ACM*, 20:832–840, 1977.

[17] I. IEC. INTERNATIONAL STANDARD ISO/IEC 11179 Information technology - Metadata registries (MDR). Technical report, ISO / IEC, 2004.

[18] D. D. Initiative. Data Documentation Initiative (DDI): Technical Specification Version 3.1. Technical report, DDI Alliance http://www.ddialliance.org/, 2009.

[19] T. Levendovszky, B. Rumpe, B. Sch tz, and J. Sprinkle. 9 model evolution and management. In H. Giese, G. Karsai, E. Lee, B. Rumpe, and B. Sch tz, editors, *Model-Based Engineering of Embedded Real-Time Systems*, volume 6100 of *Lecture Notes in Computer Science*, pages 241–270. Springer Berlin / Heidelberg, 2011. 10.1007/978-3-642-16277-0_9.

[20] Microsoft. Extensible Application Markup Language (XAML). Technical report, Microsoft, 2010.

[21] A. Narayanan, T. Levendovszky, D. Balasubramanian, and G. Karsai. Continuous migration support for domain-specific languages. In *The 9th OOPSLA Workshop on Domain-Specific Modeling*, Orlando, FL, 2009.

[22] D. M. Strong, Y. W. Lee, and R. Y. Wang. Data quality in context. *Commun. ACM*, 40:103–110, 1997.

[23] M. Vardigan, P. Heus, and W. Thomas. Data documentation initiative: Toward a standard for the social sciences. *International Journal of Digital Curation*, 3:107–113, 2008.

[24] W3C. Semantic Annotations for WSDL and XML Schema, (W3C Recommendation 28 August 2007). Technical report, W3C, 2007.

[25] W3C. XForms 1.1 (W3C Recommendation 20 October 2009). Technical report, W3C, 2009.