A Definition of Database Design Standards for Human Rights Agencies

by Patrick Ball, et al.

AMERICAN ASSOCIATION FOR THE ADVANCEMENT OF SCIENCE

1994

TABLE OF CONTENTS

1. Introduction

In 1993, HURIDOCS published their Standard Human Rights Event Formats (Dueck et al. 1993a) which describe standards for the collection and exchange of information on human rights abuses. These formats represent a major step forward in enabling human rights organizations to develop manual and computerized systems for collecting and exchanging data. The formats define common fields for collection of human rights event, victim, source, perpetrator and agency intervention data, and a common vocabulary for many of the fields in the formats, for example occupation, type of event and geographical location.

The formats are designed as a tool leading toward both manual and computerized systems of human rights violation documentation. Before organizations implement documentation systems which will meet their needs, a wide range of issues must be considered. One of these problems is the structural problems of some data having complex relations to other data.

A human rights violation can involve many victims, and one victim can be involved in many events. A large event can be made up of many smaller ones, different victims will be caught up in different actions within the event. The victim or event records themselves may contain differing degrees of repeating data, such as multiple occupations, health information, or varying locations. An example of the kind of problems which arise is that a major event may contain many violent acts, such as killings, arbitrary detention, beatings. Even if a HURIDOCS database records which victims are involved in an event, it could not simultaneously record which were victims of which specific violent acts.

Another problem faced by database designers working from the HURIDOCS formats concerns the distinction between a person and a role. There are separate formats for Victims, Sources, and Perpetrators, each of which each contain some of the same fields. Database designers need to build interfaces that help users to avoid duplicating work. The formats themselves do not solve the problem of the variety of the kinds of data that human rights organizations may want to record. For example the abuses in different regions of the world require different information to be recorded. Databases designed for the needs of different human rights agencies all wanting to be compatible with the formats would often want to use a different subset of the HURIDOCS standard fields and terminology.

Between 25-29 July, 1994, the authors of this working paper met at the AAAS headquarters in Washington, D.C., in order to establish design standards for the implementation of human rights databases compatible with the HURIDOCS formats. All of the participants had been involved in many years of work in human rights databases. We felt that by coming to agreements about how such databases ought to be structured, we could help the designers and programmers of future projects avoid the problems we had faced.

The first task of a paper-based documentation system is to decide what information needs to be collected. In contrast, the central priority of a database designer is to decide how already-collected information will be related in a complex representation. As stated earlier, we believe that our task is to suggest how to represent the complex relations in human rights data in electronic databases. More concretely, we decided that our goal was to define clear methods, or rules, for how human rights databases could represent complex links between people, organizations, events, and interventions. Therefore the central section of this paper is the definition of these rules, with kinds of information that could be represented using the rules and examples of how it might be done.

The task force's philosophy can be summed by the following maxims:

1. The first priority of any human rights database is not to introduce any additional ambiguity into the data. That is, to the extent that the original sources permit, the database must be absolutely precise regarding who committed which violations against whom.
2. The second priority of a human rights database is to be as thorough and flexible as is manageable in order to represent as wide as possible a range of abuses, interventions, people, organizations, and the complex relations that exist between all of them.
3. In order to realize the first two priorities while using the HURIDOCS formats as basic building blocks, we have tried to develop structural rules which represent the simplest, or most atomic, relations between basic entities. This means in practice that we have tried to define the most parsimonious structural rules possible, i.e., rules with the fewest roles.

The task force does not consider that the rules defined here have to be implemented on any particular platform or with any particular software package. The rule standards presented in this paper are modelling ideals, that is, techniques for representing particular formations of information necessary in the work of a human rights agency. These ideas could be implemented on a large Unix system running client-server SQL and sophisticated user software, or on a modest microcomputer using DOS and dBase.

With this document, we suggest HURIDOCS-compatible implementation standards that database designers can consider in an effort to improve the breadth and power of the information processing available to human rights agencies' work. Examples of possible xBase and SQL implementations are given in the final sections. In a supplement to this paper, there is an extensive glossary of the terms we use to define the proposed system.

The authors would like to emphasize at the outset that the material presented here is quite abstract. Our specifications deliberately leave a great deal of discretion to the designer exactly because we hope to set standards which suggest considerable rigor in the representation but do not demand that a particular kind of hardware or software be used. There is no minimum computer needed for these ideas, although as with any programming, more powerful computers will make possible more powerful analysis and a simpler interface for the user.

2. Definition of an Entity

The basic building blocks of any database system are the ENTITIES that system represents. The entities the task force isolated as fundamental to database design for human rights agencies are listed below. The entities are defined only very generally; later we will discuss each in much greater detail.

NAME - a list of all the people and organizations in the database. These may be victims, perpetrators, or information sources. A certain person may be a victim of one violent act and a source of information about other acts. Whatever their role, all the people and organizations should be maintained in a common list. [see entry in Glossary]

EVENT - the context in which the database tracks different acts (sometimes called a "case"). An event is the context or frame an agency uses to make sense of a sequence of concrete acts. An event is not an act. For example, if the agency hears that a particular person has been arrested, the arrest is an act, not an event. The arrest act would be part of the event, connected to other relevant acts. An event might be compared to a film; each frame of the film would be an act. See the Diary section, below. [see entry in Glossary]

INTERVENTION - actions taken by family members, the human rights agency, or others on behalf of victims. As with Event, interventions are composed of acts, although intervention acts are quite different from event acts. [see entry in Glossary]

These three entities (NAME, EVENT, INTERVENTION) are called BASIC entities because they are used to give form to the LINK and VOCAB entities.

DIARY - a special kind of entity used to track complex links between other entities. Diary records represent information according to the RULE-STRUCTURE defined in a RULE (see the glossary for more detailed definitions of the capitalized terms). Rules fall into one of the following three RULE-TYPEs: act, biography, or relationship. Section 6, entitled "Rule-types with rule examples and rule-instance examples," gives a detailed description of the rules and their structures. The following paragraphs are to describe the conceptual basis for each of the three rule-types. [see entry in Glossary]

ACT - a concrete, indivisible occurrence between two NAMEd entities (i.e., two people, two organizations, or a person and an organization) that happens at a specific point in time. The following instance is an example of an instance of an act-of-violence rule: On 11.08.1988, at 11 am, Lt. Mendoza executed by shooting Juan Pueblo. This act, along with others (e.g., Mendoza's arrest of Pueblo, threats Mendoza previously made against Pueblo, etc.) might be associated with an event. The human rights agency recording the information might use the event to connect individual acts into a useful, coherent form for their work. The act is something very precise which happened in the world, whereas the event is the context or frame that the agency puts on a series of acts. There are many kinds of acts -- see Section 6 for a complete list of act rules with examples. [see entry in Glossary]

BIOGRAPHY - a status experienced by a person from a starting time to an ending time. For example, the following is an example of an instance of the military-biography rule: From 01.01.1988 until 01.01.1989, Lt. Mendoza was the operations officer of the 3rd Brigade. This biography moment, along with others (e.g., other jobs Mendoza held in the military, his educational record, party militancy), is associated with Mendoza's record in the NAME database. There are many kinds of biography moments -- see Section 6 for a complete list of biography rules with examples.[see entry in Glossary]

RELATIONSHIP - a connection between two entities (two people, two organizations, or a BASIC entity with a VOCAB item) which may or may not have beginning or ending times. For example, the following is an example of an instance of the relationship-kinship rule: Juana de Pueblo has been married to Juan Pueblo since 15.05.1962. This relationship, along with others (their children, colleagues, neighbors), helps define the social context in which the agency's data exists. There are several kinds of relationships -- see Section 6 for a complete list of relationship rules with examples. [see entry in Glossary]

VOCAB - a controlled vocabulary list. Vocab items include items such as geographical terms, specific types of violence, occupations, etc. [see entry in Glossary]

Vocab is more than a controlled vocabulary list, however. The items in Vocab are organized into sets of related categories. VOCAB items also include field names and entity titles. Any item might contain a subcategory of items. Any given item plus its position in a category, and that category's position relative to other categories, is called a descriptor.

Because different human rights agencies have different needs, we build on an idea developed in the HURIDOCS formats that a given entity might have a variety of fields. That is, the field

structure of the entities is not fixed by our data model. For example, the EVENT table might or might not contain fields for information about Supporting Documents (HD#123), Date of Entry (HD#124), or Date Received (HD#125) (see Dueck et al. 1993a for the fieldname references). All three of these fieldnames would be records in the Vocab table. The only required field in any entity is its primary key, which is a code number or id field which uniquely identifies that record among all records in the database.

One of the ways that uncertain field structures can be managed at the database level is to represent information by using the fieldnames as part of the data. A complete explanation of this technique would be too extensive to present here, although it is introduced in the abstract fields and SQL implementation sections below. The basic idea is that instead of a list of fields attached to each key, the key exists in one table with "uncontrolled" information -- unvalidated strings, numbers, etc. The "controlled" information associated with the key is kept in another table (called the CV data table in the example below), linked by the entity's key. The field name which defines the controlled information is kept in one field of the CV data table. The value of the field is kept in a third field. This technique permits each field to have multiple values (by repeating the entity_keys and field_names in multiple records), and permits fields to be added or deleted without changing the database structure. This technique is the basis for the description RULE defined in Section 6.

Entity Table		**CV data table**
entity_key	----------1:m---------	entity_key
uncontrolled data		field_name
		field_value

Vocab also includes the roles that entities play vis-a-vis one another in complex links. For example, in an act-of-violence, a perpetrator does something to a victim. "Victim" and "perpetrator" are roles that particular people or organizations play in a particular act. As we discussed in the Diary section, a concrete act is represented by a link between two NAME entities. This connection will be considered at greater length in the following section, but for now note that these roles are also included in the Vocab table.

As we suggested in the introductory section, for the purpose of this paper, the actual content of the entities is not as important as how the entities relate with each other. For our purposes, it is only necessary that each record in each entity (e.g., each person or organization recorded in NAME) be uniquely identifiable by some code, called a primary key. We leave all other discussion of the content of the entities to other materials, specifically Dueck et al. 1993a.

3. Entity versus Role versus Link

Entities must be distinguished from the roles that those entities might play in any concrete act that the database is tracking. For example, in a human rights database, people are entities. Terms, including perpetrator, victims, witness, and lawyer, do not define entities but roles that persons play in particular acts. More generally, the same person might act many roles. This logic means that the concept to be represented as an entity is the person rather than the role the person might be playing in a particular situation. In many human rights information systems, database designers have defined tables to represent particular roles. This kind of representation makes sense in a paper system because different kinds of information are collected for perpetrators and victims, for example, even though both are usually people. However, the difference between a perpetrator and a witness, in database terms, is the different roles they play in particular acts. The

task force suggests that these roles be implemented in links between entities, and not by having different tables for different roles.

A rule defines the association between the basic data entities which a HURIDOCS compatible database may need to support. For example if a human rights "act of violence" is to be recorded, a victim, a perpetrator, a date, an event type and a location should be expected. Any one of these items may be unknown, but they are all expected for a complete record. Including "acts," we identified three fundamental types of rule, outlined below.

Structural rules define how tables connect together. Tables link by "one to many" (in shorthand, 1:m) and "many to many" (m:m) techniques to which are attached substantive meaning. The RULEs presented in Section 6 are of this type.
Scope rules define hierarchic relations, for example between events and acts, or between large, abstract events, and smaller events.
Sequence rules define how to represent sequences of related entities, for example an arrest followed by a trial, sentencing, imprisonment, appeal etc.

We developed three types of structural rule: act, biography, and relationship. The definitions are presented in Section 6.

The scope rules we agreed on was only the rudimentary recognition that EVENTs are the conceptual creations of human rights agencies used to order the vast array of acts which compose a human rights violation. Thus EVENTs are composed of acts. Also, we agreed that some events may be very large, and thereby encompass smaller events. For example, imagine event such as the story of the kidnapping and murder of three men from a family ("Disappearance of the Pueblo brothers"). This event might be composed of a large number of specific acts (the kidnappings, acts of torture, the killings, exhumations, etc.). However, the agency recording this event might classify it as part of a trend they are tracking, say "The scorched-earth campaign of Col. Alvarez." The campaign would be an event which would be associated with the various smaller, more specific events such as the Pueblo brothers kidnappings.

Sequence rules must still be developed.

4. What is a Link?

Data that are acts, biography moments, or relationships are stored as links between records in the name, event, intervention, and vocab tables. How the link is represented in the actual database depends on the implementation of the rules (see the implementation sections for examples). In this section, the idea that the name, event, intervention, and vocab elements are connected together to form new data structures will be explored in more detail.

No matter why or how the agency knows a certain person, in this data model that person will be represented by a single record in the NAME table. If that person is the victim of a particular act (back to Juan Pueblo who was shot by Lt. Mendoza), the KEY (see glossary) for Juan Pueblo will be associated in a link, called a rule-instance, with the key for the idea "executed by shooting" and with the key for Lt. Mendoza. The keys are associated in such a way that Pueblo's key is in the "victim" position, the "executed by shooting" is in the "action" position, and Mendoza's key is in the "perpetrator" position. The positions define roles in the rule-instance.

Information stored in this way is very flexible. If fully implemented, the link table enables the full viewing of data from any angle, so that, for example, the user can trace all the people

thought to be victims of a given perpetrator; alternatively, if a victim has been the subject of several violations, all the perpetrators of that victim can be seen from the victim's record. The link table assigns roles to the actors in a link, so if the data is viewed from the point of view of the perpetrator it is clear that the perpetrator is the subject, the act the verb and the victim the object.

If a person has multiple roles in an event, or is involved in different roles in different events, all of that information is available either via the person's record (which would show all the acts in which this person was involved in any role), or by each of the events in which the individual acts were classified by the agency.

5. Presentation of a Data Model

The links defined by the rules below connect entities in particular, structured ways (see Figure 1). In database terms, each record in the entity tables can connect to 0, 1, or many records in the link table.

The entity tables are on the outside of the diagram (Name, Intervention, Vocab, Event, and Diary). The LINK table is in the center. Note that each of the lines connecting an entity table to the link table ends at the link table with a zero and a "crow's foot" branch. The line defines a 1:m relationship between the ENTITY tables and the LINK table.

The Name, Event, and Intervention tables are called BASIC tables because they will give meaning and context to the rule-instance records the user builds. Vocab contains the controlled vocabulary items and the category structure which organizes the items.

The Diary-Link structure may be implemented slightly differently in different versions. In a standard relational model (see Section 8.1), Diary and Link may be embedded in the same tables. In abstract or more highly relational models (Sections 8.2, 9), the Diary table contains the master records for the rule-instances; the actual data for the rule-instances is kept in the Link table.

As stated earlier, the Data Model proposed by this working group does not specify exactly how the link works in any given implementation of these standards. The link structure might be a collection of standard relational tables, or it might use a more sophisticated, more highly normalized technique. At a minimum, for a given implementation to be compatible with this standard, we specify that it must be able to represent information in the form of the three rule-types, and be able to manage all of the rules proposed in the following section. It should be emphasized that these are minimum standards -- much greater detail and analytical power are possible.

6. Rule-types with Rule Examples and Instance Examples

The working group proposes that three basic rule-types cover the majority of the complex relationships between names, events, interventions, and vocabulary items. The three types (act, biography, relationship) are defined according to the following form, called the RULE-

STRUCTURE:

i. Rule-Type Name - indicates the fundamental nature of the rule, such as the type of event it describes

ii. Rule-Type Definition

iii. List of Linked Entities

iv. Description of its Time-Structure - i.e. how many dates are needed to describe it

v. List of Vocab Types and Categories - which might qualify the linked entities' relationship.

vi. The Presence or Absence of a Notes Field for Additional Description

vii. Example Rules with (8.1) Explanation of the Rules

viii. Typical Narrative - with a table-like rendering of the narrative according to the appropriate rules. Note that the rendering of the data in the narrative more or less follows the structure suggested in the Standard Relational Model. However, how the data are actually represented in the database depends on the specific implementation chosen. See the sections 8.2. Abstract Fields and 9. SQL for other possible techniques.

6.1 Type of rule: Act

---

i. Type of Rule: Act

ii. Definition - a single, indivisible act involving 2 persons or organizations in the context of an event.

iii. Links

a. Subject: the subject of an act is the person or organization who initiated the action. The subject could be a specified or unknown person, organization, or class of people.

b. Object: the object of an act is the person or organization who received the action.

c. Event: the event defines the context of other acts, interventions, or ramifications of the act in which this act occurs.

iv. Time Structure - an act occurs at a single point in time.

v. Vocabulary Items

a. Action: the action defines exactly what was committed by the subject against the object.

b. Location: the location defines exactly where the act occurred.

vi. There is a Notes Field

vii. Example Rules

RULE	ROLES			Descriptors			
Example	**Subject**	**Object**	**Context**	**Action**	**Location**	**Date**	**Notes**
a. violence	perp.	victim	event	action	place	day/time	"
b. legal norm violated	perp.	victim	event	norm violated	place	day/time	"
c. facts of the case	submitter of fact	accused	event accused	nature of fact	place	date of accusation	"
d. legal decisions	legal org'n	accused	event	result	place	date of ruling	"

vii(1) Example Discussion

a. An act of violence could be of many types. It could be an act of torture, an arrest, or a killing. The key to these acts is that the action committed is violent, distinguishing this kind of act from other acts. Possible violent actions would include items chosen from HURIDOCS Supporting Documents J, Type of Event (Dueck and Noval et al. 1993b).

b. Legal norm violations represent the same acts as acts of violence but from a legal, rather than from a narrative perspective. Acts of violence describe occurrences in the world by a literal explanation ("Lt. Mendoza executed by shooting Juan Pueblo"). Legal norm violations describe the same occurrences by an interpretation in legal terms ("Lt. Mendoza violated the National Constitution and the Universal Declaration of Human Rights against Juan Pueblo"). The CATEGORY of the verbs for legal norm violations would include a list of the national, regional, and international laws, conventions, and protocols which govern the rights of citizens vis-a-vis their governments.

c. The facts of the case include complaints, investigations, charges, and defense motions, among other legal definitions of fact.

d. Among the kinds of legal decisions an agency might want to represent could be judgements, sentences, or appeal decisions.

viii. Act-of-Violence and Act-Legal-Norm Instance Examples

The victim (V0050290) was arrested in Ramallah on 1 December 1989 and taken to al-Moscobiya detention center where he was severely beaten, deprived of sleep and sufficient food, choked to the point of losing consciousness, and subjected to the "shabeh," whereby the prisoner is forced to stand for prolonged periods in the open, with his head covered and hands tied behind his back, exposed to all weather conditions. Ten days later (i.e., 11 Dec 89) he was transferred to Ramallah prison, where some of this treatment continued. Whilst there, his arm was apparently broken by a Shabak (intelligence) officer known as "Max" (P0502901), although it was only two weeks later that he was taken to a hospital and his arm put in a cast. Following an extension of his detention order by a judge on 17 December, he was returned to Moscobiya prison. He began a hunger strike on 7 Jan 90. (Dueck et al. 1993a:29, ff.)

Notes*: all the acts represented here were classified by the agency receiving the denouncement as relevant to the same event (E005029); there is only one victim (V0050290). Two kinds of acts are rendered: a) acts of violence, b) legal norm violations. This is only an example; an actual agency would, for example, include the legal norm violations for its country's constitutional*

protections. The codes for locations, perpetrators, and actions all follow Dueck and Noval et al. (1993b), and are presented here for clarity:

```
26.1      al-Moscobiya detention center
26.2      Ramallah prison
40        intelligence service
60        judiciary
05.211    Slapping, kicking, or punching
05.261    strangling
05.272    "planton" or forced standing
05.65     bound
05.41     deprived of food
05.43     deprived of sleep
05.2141   breaking bones - arm
05.44     deprived of medical attention
04.071    extension of administrative detention
D3        Convention against torture (1984)
A2        International Covenant on Civil and Political Rights (1966)
```

rule	subject	action	object	event	loc.	date	notes
a.	60	05.211	V0050290	E005029	26.1	19891201	
a.	60	05.261	V0050290	E005029	26.1	19891201	
a.	60	05.272	V0050290	E005029	26.1	19891201	
a.	60	05.65	V0050290	E005029	26.1	19891201	
a.	60	05.41	V0050290	E005029	26.1		
a.	60	05.43	V0050290	E005029	26.1		
a.	P0502901	05.2141	V0050290	E005029	26.2		
a.	60	05.44	V0050290	E005029	26.2	19891215	
a.	40	04.071	V0050290	E005029	26.1	19891217	
b.	60	D3	V0050290	E005029			torture
b.	40	A2	V0050290	E005029			due process

6.2 Type of rule: Relationship

---

i. Type of Rule: Relationship

ii. Definition - relationships between individuals or organizations when no act is involved.

iii. Links - 2

First person or organization: one party to the relationship

Second person: the other party to the relationship

iv. Time Structure - initial and final dates, plus a date which defines as of when the agency became aware of the relationship

v. Descriptors

Nature of relationship: the quality of the connection between the two parties

vi. Notes - there are notes associated with relationship

vii. Example Rules

	ROLES		DESCRIPTOR	
Example	**party**	**party**	**quality**	**date**
a. family	relative	relative	kinship tie	"
b. other personal	first person	second person	quality of connection	"
c. organizational: hierarchical	super-org.	sub-org.	quality of connection	"
d. organizational: affiliational	first org.	second org.	quality of connection	"
e. event-scoping	super-event	sub-event	reason for connection	"
f. description	BASIC	VOCAB	field-name	"

vii(1) Example Discussion:

a, b. Kinship relations are often quite important to understand why a pattern of violations has occurred. Other relations (friends, neighbors) can be represented in a very similar way.

c, d. Organizations are related to each other in a variety of complex ways. Hierarchical organizations indicate organizations that are part of larger organizations (e.g., a certain brigade pertains to a certain division; a trade union belongs to a federation of trade unions). Affiliations between organizations indicate simple connections (e.g., a trade union may be affiliated with a political party).

e. as described in previous sections.

f. BASIC entities (NAME, EVENT, INTERVENTION) which have controlled vocabulary attributes (e.g., occupation, location) store those attributes as RULE-INSTANCEs of the Description RULE.

viii. Typical Narrative - with a rendering of the narrative according to the appropriate rules.

Juan (P1) Pueblo, a farmer, married Juana Castaneda (P2) in 1962. Juan's father was Jorge Pueblo (P3).

The International Confederation of Free Trade Unions (ICFTU) -affiliated Norwegian Labor Confederation (LONOR) had been cooperating with the Salvadoran National Federation of Trade Unions (FENASTRAS) for 6 years (date of report: 1989). The 31.10.1989 bombing of the FENASTRAS Headquarters in San Salvador is widely seen as the trigger for the November, 1989, FMLN General Offensive. Several of the victims of the FENASTRAS bombing were members of the FENASTRAS member union SOICES. (adapted from Dueck et al., 1993a, pp. 41-4)

rule	**party**	**party**	**quality**	**date**	**notes**
a	P1	P2	married	00.00.1962	
a	P3	P1	father-child		
f	P1	farmer	occupation		
c	ICFTU	LONOR	int'l confederation		

c	FENASTRAS	SOICES	nat'l confederation		
d	LONOR	FENASTRAS	int'l cooperation	00.00.1983	
e	Nov.Offensive	F.Bombing	trigger event		

6.3 Rule type name: Biography

i. Rule-Type Name: Biography

ii. Rule-Type Definition - a time-bounded status experienced by a person.

iii. List of Linked Entities

1. Person - the person to whom this status pertains.

2. Organization - or what confers the status: the organization (if any) in which the person experiences this status

Descriptors:

iv. Time - beginning and ending dates, as well as when the status became known to the organization

v. Descriptors - that qualify the linked relationships

1. Situation - the job the person hold, or the reason the person is related to the organization.

2. Rank - the level a person has in this organization.

vi. There are Notes kept with Biography Records

vii. Example Rules with (8.1) Explanation of the Rules

		LINKS		DESCRIPTORS			
Rule	**name**	**person**	**what confers the status**	**situation**	**level**	**initial, final, & as of dates**	**notes**
a.	military	soldier	unit	post	rank	"	"
b.	government	official	ministry	job	rank	"	"
c.	ONG's	activist	organization	job	--	"	"
d.	party	militant	party	role	--	"	"
e.	education	student	school	student	level	"	"
f.	health record	patient	health center	illness	--	"	"
g.	status of dspd	victim	susp. perp.	dspd	--	"	"
h.	detention	detainee	detaining organization	detention status	--	"	"

viii. Typical Narrative - with a rendering of the narrative according to the appropriate rules.

Pablo Mendoza (P5) graduated from the military academy in 1961. He was assigned to the 3rd Brigade as a 2nd lieut., platoon commander, from July, 1961 until July, 1962, when he was promoted to 1st lieut. He was then transferred to the General Headquarters where he worked a staff position in logistics until December, 1964. Promoted to captain, he was made commander of the Jaguar Battalion of the 2nd Brigade, which he remained until his accident on 16 March 1966. After 8 months in the Military Hospital he was well enough to be discharged.

Juan Pueblo (P1) joined the XYZ Party in 1971; he remained active until 1975. On 25 July 1975 he was detained by the National Police. Although he was released on 15 August, he did not return from work on 17 August. He has not been heard from since.

Rule	person	status	situation	level	initial date	final date	notes
e	P5	mil.ac	student	grad	19610000		
a	P5	3rdB	PltCdr	2ndL	19610700	19620700	
a	P5	GHQ	Log	1stL	19620700	19641200	
a	P5	2ndB	BatCdr	Cap	19641200	19660316	Jaguar
f	P5	MilHsp	back prob	-	19660316	19661116	
d	P1	XYZ	militant	-	19710000	19750000	
h	P1	NatPol	detained	-	19750725	19750815	
h	P1	-	dspd	-	19750817		

7. Rule Parsimony -- the trade between precision and simplicity

It is possible to track human rights violations very precisely using rules defined as simply as above. There is, however, a trade between defining very simple rules, and defining rules that reflect an agency's specific needs and the specific forms of information that they receive. The simpler the rules are, the less chance there is that an agency will inadvertently introduce ambiguity into their data. However, the more elaborated the rule definitions, the more fully the agency can represent information of particular interest to them.

The rules we elaborated in section 6 are the minimum set of rules that a HURIDOCS compatible database must represent. The abstract field and SQL implementation examples are designed in order to permit very precise and thorough customization of the rules such that each agency can represent information in a form very close to how they receive and understand that information. The standard relational model, on the other hand, is designed only to represent the simple rules defined above. It would be adequate for many agencies' needs -- especially for those agencies that do not have extensive resources to devote to computing personnel and hardware.

One way to balance the goal of simple rules with the goal of very thorough rules is to build elaborate controlled vocabulary lists. For example, consider the following narrative. "Juan Pueblo was arrested by the National Police on July 19, 1979, and taken to their headquarters. While there, they tortured him by blows to the head and thorax, and by immersing him in contaminated water, and by giving him electric shocks to the thorax with an apparatus."

In the standard representation described above, all of the tortures the National Police subjected Juan Pueblo to can be classified by the act-of-violence rule -- but we will need a list of controlled vocabulary items (in the "action" column) which includes many items.

rl	sbj	action	obj	event	loc	date	notes
a.	NP	arbitrary arrest	JP	E01	NP-HQ	19790719	
a.	NP	blows-head	JP	E01	NP-HQ	19790719	
a.	NP	blows-thorax	JP	E01	NP-HQ	19790719	
a.	NP	drown-cont.water	JP	E01	NP-HQ	19790719	
a.	NP	elctc-apparatus	JP	E01	NP-HQ	19790719	

However, some agencies might want to design slightly different rules for all of these different kinds of tortures. Below there are three different rules to represent different kinds of torture.

rl	sbj	action	obj	event	loc	date	notes
a.	NP	arbitrary arrest	JP	E01	NP-HQ	19790719	

rl	sbj	action	body part	obj	event	loc	date	notes
b.	NP	blows	head	JP	E01	NP-HQ	19790719	
b.	NP	blows	thorax	JP	E01	NP-HQ	19790719	

rl	sbj	action	body part	technique	obj		loc	date	notes
c.	NP	e.shock	thorax	apparatus	JP	E01	NP-HQ	19790719	

In the first example, there is ONE rule type for ALL acts of violence. The rules are simple, but the controlled vocabulary list is more complicated. In the second example, the rules are more complicated, but the controlled vocabulary lists are simpler. One advantage of the more complicated rule system is that the rules reflect even more closely what the narrative describes. In systems that store rule information as "meta-data," such as the abstract field and SQL models presented below, there is little penalty for adding rules. But in the standard relational model, it is difficult to handle more than a few RULE-TYPEs.

So far, we've discussed these rules only as they are classified according to structure. To users, however, rules make sense in substantive terms: rules about violence are together, rules about legal process are together, etc. Organizing rules by structure is not always quite the same as organizing rules by meaning. It is the job of the programmer to build an interface which organizes all the information according to whatever makes most sense to the user. A good interface will be the difference between an incomprehensible system and a friendly one.

Notice that systems defined as "HURIDOCS compatible" are NOT "compatible" with each other in the sense that people can simply trade disks and use each other's data. The very structures of the tables may well be different between two compatible systems. Any two systems which want to share information directly need to establish an exchange format, including field structures, file organization (DBF, ascii, etc.), and most importantly, a common organization of controlled vocabulary items.

8. xBase Table Design Implementation Example

In this section, we will present two examples of database implementations of the data model and rules we have discussed in this working paper. The first example uses a standard relational design, the second uses "abstract fields," that is, a much more normalized design which permits greater design flexibility and increased data representation precision.

Both examples are designed for the "xBase" family of microcomputer database management systems (DBMS) for DOS, Windows, or Macintosh. dBase (III, III+, IV, IV-Win), Fox (Base, Pro v.2 and following), Clipper, Quicksilver, or even Paradox would be included in this group, as would other fully relational DBMS packages. It would be considerably more difficult to implement these examples in pseudorelational systems, such as the ISIS family (CDS, Mini- or Micro-). The hardware platform depends on the demands that the DBMS makes, but we recommend a 386 or greater with RAM sufficient to support the DBMS (e.g., 4 mB under DOS, 8+ mB under Windows).

8.1 xBase Implementation Example 1: The Standard Relational Model

---

The Standard Relational Model (SRM) is presented in diagram form in Figure 2. As with the first data model, the SRM begins with the four basic tables: the entities (NAME, EVENT, INTVN) and the descriptor table VOCAB).

The SRM differs from the abstract fields and SQL models principally in its handling of the LINK structure. Optimally, all the data in the LINK structure should be in a single table so that all the links from any chosen BASIC entity are immediately viewable -- that is, the user can view all the different ROLEs this entity plays in all the different RULE-INSTANCEs to which it connects. In the SRM, the user can see all the links from any given BASIC entity, but only one RULE-TYPE at a time. That is, the user can view the links from a BASIC entity to other entities of the Act RULE-TYPE, or of the Biography or Relation RULE-TYPE, but only one at a time. This limitation arises because the RULE-INSTANCES of different types are in different tables. INSTANCES of RULES of the ACT RULE-TYPE, for example, are in the ACT table, whereas INSTANCES of the BIO RULE-TYPE are in the BIO table.

This model maps the RULE-STRUCTUREs directly onto three tables, one table each for act, biography, and relationship. The structure of each RULE-TYPE is reflected exactly in the field structure of each table. Because the fields are fixed in this model, each RULE-INSTANCE can accept only one value for each of the ROLEs specified.

For example, in a given act-of-violence rule-instance, there can be only one victim, one violent act, one perpetrator, and one event per rule-instance. If the narrative of the event includes acts-of-violence in which there is, for example, more than one victim, the user must create one rule-instance for each victim. The same is true if there is more than one violent act, more than one perpetrator, etc. This is not a limitation of more sophisticated models.

Note that in the SRM, each RULE-INSTANCE is represented by exactly one record. Further, each ROLE is represented by a FIELD. This is intuitively appealing, and for simpler implementations the SRM can be entirely adequate.

The SRM has several advantages. All implementations of this standard have to maintain a considerable amount of "meta-data," that is, definitions of data separate from the actual table structures. The SRM minimizes meta-data. The RULE-STRUCTURES, at least, are already encoded in the table structures. There are still aspects that need to be defined for each rule: which VOCAB categories fit in which roles, and which entities belong in which roles, still must be defined in program code.

Any implementation of the SRM will require a great deal of tricky user interface programming to assist the user's data entry. All of these models force the user to enter a lot of data -- any tools that we can provide to make his or her job easier will be very useful.

8.2 Abstract Fields and Dynamic Data Types

Discussion in this section will draw heavily from theory presented more fully in Ball (1993).

This implementation has two principal objectives: first, to implement all the different RULE-TYPEs in a single table, and second, to allow the user to view simultaneously all the RULE-INSTANCEs in which a given BASIC entity is involved. Secondary benefits include the ability to include multiple BASIC links in each ROLE. Considering the example above, in which an instance of the act-of-violence rule had more than one perpetrator. In this situation, multiple perpetrators can be attached to the

Consider the following data from the data rendering examples.

ACT						
rule	**subject**	**action**	**object**	**event**	**loc.**	**date**
a.	60	05.211	V0050290	E005029	26.1	19891201

RELATION					
rule	**party**	**party**	**quality**	**date**	**notes**
a	P1	P2	married	19620000	

BIOGRAPHY							
Rule	**person**	**status**	**situation**	**level**	**date**	**date**	**notes**
e	P5	mil.ac	student	grad	19610000		

in the abstract fields model, the information above would be represented in two related tables, LINK and DIARY. The information in the LINK structure would appear as follows:

LINK			
diary_id	**role**	**data_id**	**table_ptr**
001	sub	60	name
001	action	05.211	vocab
001	obj	V0050290	name
001	event	E005029	event
001	loc	26.1	vocab
001	date	19891201	date
002	fparty	P1	name
002	sparty	P2	name
002	quality	married	vocab
002	date	19620000	date
003	person	P5	name

003	status	mil.ac	vocab
003	sit'n	student	vocab
003	level	grad	vocab
003	date	19610000	date

LINK need not be human readable because what a person needs to make sense of in a rule-instance is the entire rule-instance -- not a single role in the instance. A human-readable summary of the instance is contained in Diary.title_info. By storing the LINK as compressed binary key information, the LINK record can be reduced to about 12 bytes. The size of the LINK record is important because there will be very many of them. In the diary table, the following would appear:

DIARY			
diary_id	**rule_type**	**title_info**	**notes**
001	act-of-violence	victim severely beaten	
002	relation-family	P1 and P2 marry	
003	bio-education	P5 graduates from the military academy	

As in LINK, the diary_id and rule_type fields can be stored in internal binary forms. Each RULE-INSTANCE in the abstract model is represented by one DIARY record and an indefinite number of records in LINK. All the LINK records that pertain to a single RULE-INSTANCE are related to the master record for the rule instance in DIARY by common key values in diary_id. All the BASIC entity key information is contained in a single field, LINK.data_id. DIARY.title_info contains a human readable form of the information stored in the collection of records in LINK that are associated with this DIARY record.

Given this structure, it is easy to browse all the links associated with a given BASIC entity record in xBase. The following code is from FoxPro, but the logic would be similar in compatible DBMS'. Assuming that we are in the work area of the entity we want to view, and the record pointer is positioned at the record we want (NB: '&&' indicates a comment in Fox):

```
select DIARY
set order to diary_id
select LINK
set order to data_id
set relation LINK.diary_id into DIARY      && LINK   --m:1--> DIARY
select <BASIC entity>
set relation <BASIC>.data_id into LINK     && BASIC --1:m--> LINK
select DIARY
browse fields DIARY.title_info             && BASIC -- 1:m --> DIARY
```

Any given BASIC record yields m Diary records. The Diary browse shows only those records associated by some link to the record in the BASIC entity we're analyzing. This can be a very powerful analytical tool, and runs quickly even with very large data files.

The most difficult part of managing an implementation of this model is the meta-data. In other words, how should the system manage the RULE definitions? The most obvious solution is to store rule definitions in program code. There are a variety of possibilities which afford greater flexibility, for example, the solution presented below, in Section 9.3. See Ball (1993) for an

xBase implementation of the same ideas. A fuller explanation of the ideas would be beyond the scope of this paper.

9. SQL Database Design Implementation Example

```
                        >-----Name  -----<
            Link        >-----Intvn -----<     Term  >----- Vocab
                        >-----Event -----<
                        >-----Diary -----<
```

Table Fields:

```
NAME TABLE:
     name_id
     full name
     sort name
     dob

EVENT TABLE:
     event_id
     event_name

INTVN TABLE:
     intvn_id
     intvn_name

TERM TABLE:
     entity_id      (points to Diary, Event, Intvn or Name record)
     table_ptr      (points to Diary, Event, Intvn or Name Table)
     vocab_id

LINK TABLE:
     diary_id
     entity_id      (points to Event, Intvn or Name record)
     table_ptr      (points to Event, Intervn or Name table)
     role

DAIRY TABLE:
     diary_id
     rule_name
     start_date
     end_date
     other_date
     diary_notes

VOCAB TABLE:
    vocab_id
    role
    term
```

The NAME, INTVN and EVENT tables represent the BASIC entities. Most of the attributes in these tables are held in the TERM TABLE which consists of three fields, entity_id, table_ptr and vocab_id. For example the NAME TABLE would contain certain fields for data which was always required (eg Name, date-of-birth), but the other entity fields would be held in TERM (eg a if the vocab_id of the term 'doctor' was 6, and the name_id of a person on the database who was a doctor was 5, this person would have a term record: 5, 'name', 6.

The LINK TABLE structure chosen is similar to that used in the second sample table design for an xBase implementation. The LINK TABLE has four fields, a diary-id field which contains the

same value for all the related records in the link (rule instance), an entity_id field containing an id value from one of the basic data tables. In addition there are two further fields for specifying the entity-id, a table_ptr field which indicates the id type (ie. which table it refers to) and a role field which indicates the role of the table-id in the rule, for example a person from the name table can be in the role of victim, perpetrator or source.

The DIARY provides a simple and very flexible structure for recording information as it is acquired and reflecting time sequences in the data. It contains a diary_id, rule_name, a series of dates and a notes field. Other terms associated with the diary entry are stored in the TERM TABLE. In a sense it provides a fuller explanation of a link, such as dates of the act, and detailed notes about it. Depending on how the data is selected in a query, it can show all the known information about a particular case, event, source etc, or a subset, such as only the acts, or only the interventions involving a victim. It lends itself to implementation of easy, intuitive user interface tools where the user enters data as they receive it, but can then use it to analyze the complex relations in the database by choosing the different views.

9.1: Querying & Performance

This data model will result in very large link and term tables. The performance of the database is crucial, especially on queries. There are essentially two types of SQL query demanded by the model, selects involving normal joins, and nested selects. Although we have not tested all performance ramifications, we suspect that nested selects can cause performance problems on some SQL databases. We are still testing to determine how much performance degrades with nested selects. (see Ganski and Wong (1987) and Won (1982)).

An example of normal join to find all the diary records for person X in role of victim

Query 1

```
Select diary.start_date, diary.end_date, diary.diary_verb,
diary.diary_text
from name, link, diary
where link.diary_id = diary.diary_id
and link.id = name.name_id
and link.table_ptr = 'name'
and link.role = 'victim'
and name.name_id =X
```

Example of nested select: Find all the people cited as perpetrators in EVENT Y

Query 2

```
Select name.fullname
from name,link
where link.entity_id = name.name_id
and link.table_ptr = 'name'
and link.role = 'perpetrator'
and link.diary_id =
(select link.diary_id
from link
where link.entity_id = Y
and link.table_ptr = 'event')
```

It would be possible to specify a join rather than a nested select by aliasing the link table, and joining it to itself:

Query 3

```
Select distinct name.fullname
from name, link link1, link link2
where link1.id = name.name_id
and link1.table_ptr = 'name'
and link1.role = 'perpetrator'
and link1.diary_id = link2.diary_id
and link2.entity_id = Y
and link2.table_ptr = 'event'
```

However this would not necessarily improve performance as the 'distinct' keyword tends also to reduce performance. It is necessary to specify, 'distinct', as otherwise the query will return duplicate rows when a perpetrator is linked to the same event more than once.

9.2 Main Differences of the SQL Model

The main differences between the proposed SQL model and the abstract model are:

1. the term table does not link to the link table. This is order to reduce any potential performance problems with using nested selects. By separating out the term table from the link the basic attributes of the entities can be selected using simple tables joins rather than nested selects.

For example to find all the people with the occupation 'doctor'.

Query 4

```
select name.fullname
from name, term, vocab
where name.name_id = term.entity_id
and term.table_ptr = 'name'
and term.term_id  = vocab.vocab_id
and vocab.term = 'doctor'
```

If we had the terms in the link table as the abstract model does, we would have to use a nested query.

Query 5

```
select name.fullname
from name, link
where link.entity_id = name.name_id
and link.table_ptr = 'name'
link.diary_id =
(select link.diary_id
from link, vocab
where link.link.entity_id = vocab.vocab_id
and link.table_ptr = 'vocab'
and vocab.value = 'doctor')
```

2. the Diary table has a one to many relationship with the link table, however unlike the entity tables every link record must point to a diary record. This also has performance advantages as it

means the link_id can actually be the diary_id. This means no nested selects are necessary to get the diary information about a link (see Query 1).

3. All the mass events are entered in the event table. However 'indivisible acts' which are described by one rule, such as 'act of violence' are only entered in the link and diary table.

9.3 Storage of the Rules in the SQL Model

In the SQL Model the rules are stored in the database in three tables:

```
Rule_type  ----< Rule -----< Role
```

Rule_type field and simplified example data for the rule 'torture':

rule_type **act of violence**

Rule definition:					
rule_name	**rule_type**	**date1**	**date2**	**date3**	**notes**
torture	act-of-violence	date	n/a	n/a	yes

Role definitions						
row#	**rule_name**	**table_ptr**	**role**	**min_no**	**max_no**	**controlled_vocab**
1	torture	vocab	torture method	1	10	yes
2	torture	name	victim	1	1	no
3	torture	name	perpetrator	1	1	no
4	torture	event	event name	1	1	no
5	torture	vocab	location	1	1	no

* the role field stores the name that will appear on the screen to prompt for the appropriate data entry. It must also be a term in the vocab table. If the field is to be filled from a controlled vocabulary it must be a category term in the vocab table.

** the min_no specifies if the field is mandatory (0 if optional, 1 if mandatory), the max_no shows how many times this field can be repeated in the rule.

A rule instance for the above torture rule would have three rows in the link table, two pointing to the name table and one to the event table. It would also have several rows in the term table, one for the location term and between one and ten rows, one for each of the torture methods used.

References

Ball, Patrick. 1993. "Use Abstract Fields and Dynamic Data Types." FoxTalk. (December):3-13.

Dueck, Judith, et al. 1993a. HURIDOCS Standard Formats: A Tool for Documenting Human Rights Violations. Oslo:HURIDOCS.

Dueck, Judith, Aida Maria Noval, et al. 1993b. HURIDOCS Standard Formats: Supporting Documents. Oslo: HURIDOCS.

Ganski, R.A. and H. K. T. Wong. 1987. "Optimization of Nested SQL Queries Revisited", ACM SIGMOD Conference 1987, pp 23-33.

Shasha, D.E. 1992. Database Tuning, a principled approach. New Jersey: Prentice Hall, New Jersey. (ISBN 0-13-205246-6)

Won Kim. 1982. "On Optimizing an SQL-like Nested Query", Transactions on Database Systems. 7(3):443-469 (September).