A General Temporal Data Model and the Structured Population Event History Register

Samuel J. Clark*

Abstract

Longitudinal projects collect, store and manipulate large amounts of data that describe the histories of individual people, households and other units of analysis. These are *temporal* data that often describe inter-related histories, and consequently the *structure* of such data is complex and managing them can be difficult. Several existing data models successfully address this challenge but with significantly different solutions, and as a result, data stored using these different data models are hard to compare or merge. Moreover when ongoing projects use different data models, it is difficult to design an investigation that utilizes or collects data from more than one project because their individual data models are largely incompatible. Multi-site longitudinal investigations including large scale vaccine and behavioral intervention trials are and will become more common in the future making it an urgent matter to develop a standard temporal framework to guide the storage and manipulation of complex temporal data describing the histories of people (and households and other aggregations of people) living in multiple populations. This work begins to address this challenge by presenting 1) an abstract temporal data model that can represent an arbitrary range of inter-related temporal trajectories – the General Temporal Data Model or GTDM, 2) a relational database implementation of the GTDM that is able to store an arbitrary range of inter-related temporal trajectories with a single *static* relational schema – the Structured Population Event History Register or SPEHR, and 3) a relational database schema based on SPEHR that can store the contents of many SPHER-based databases allowing data from different longitudinal projects to be easily merged and managed together. The aim is threefold: 1) to improve the efficiency and effectiveness of data management at individual longitudinal projects, 2) to allow multiple longitudinal projects to easily share and compare data and implement multi-site investigations that make full use of both existing data and new data to be collected during the course of the new investigation, and 3) by providing a standardized data model, to allow the development of standardized and largely automated data manipulation, extraction and analysis tools.

(328 words)

Key Words

Data, Longitudinal, Temporal, Data Model, Database, Population Register, Relational, Event, Influence, State, Metadata, DSS, Surveillance, Methods, SPEHR.

* Institute of Behavioral Science (IBS), University of Colorado at Boulder; Agincourt Health and Population Research Unit, School of Public Health, University of the Witwatersrand; Graduate Group in Demography, University of Pennsylvania. Corresponding Addresses: P.O. Box 1773, Westville 3630, South Africa; sam@samclark.net.

Table of Contents

1 Background and Motivation

At the core of many questions that interest us is an investigation of cause and effect, or causality. Loosely, causality involves a notion of ‘consequence’ so that if **A** occurs at time *t* or pertains for times *t-n* → *t* then *as a consequence* **B** occurs at time *t+n* or pertains for times *t* → *t+n*. Time plays a central role in determining the proper sequence of these events or states so that it is possible through repeated measurement to infer that **B** results from **A** with a specified degree of certainty. Measurements of this type are inherently *temporal* (relating to time) because they must record the temporal sequence of **A** and **B**, and beyond that it is usually important to know the duration of time between **A** and **B** as well. Given the central role of time in the concept of causality, strong inferences about cause and effect can only be made from observations that span durations of time sufficiently long to observe the *cause* and the *effect*, preferably many times among a large population of whatever the unit of analysis is.

The questions that interest many population and health scientists are becoming increasingly complex and require longer periods of more intense observation to adequately describe the cause and effect of interest. The result is a rapidly increasing number of long term longitudinal projects that monitor relatively large populations of human beings. Evidence of this proliferation is the INDEPTH network (INDEPTH Network, 2004b) that brings together Demographic Surveillance System (DSS) sites located in Africa, Asia and Latin America. A large fraction of the 34 sites that are current members of the network were initiated within the past ten years.

More specifically in the era of HIV/AIDS and increased interest in malaria, there are a growing number of both biomedical and behavioral interventions being developed for implementation in developing countries (See for example: HVTN, 2004; IAVI, 2004; SAAVI, 2004), all of which need testing and eventually certification on relatively large populations of human beings. Given the complexity of some of these and the fact that they will include multiple combined “interventions” in a single delivery strategy, the study designs involved will be complex, require large numbers of participants from at least several different populations and span significant periods of time. All of this will require accurate, flexible, user-friendly data management of large amounts of complex temporal data, thereby adding both weight and urgency to the development of temporal data management standards able to support this type of data.

Longitudinal projects (such as a DSS site) collect, store and manipulate large amounts of data that describe the histories of individual people, households and other units of analysis. These are temporal data that describe the complex inter-related histories of large numbers of people and aggregations of people (for example marital unions, households and villages). The inter-relatedness of these histories is manifested in myriad relationships that people share among themselves and between themselves and various aggregations of people; these relationships can be thought of as the *structure* of the population. As a result a longitudinal project collects and manages data that describe a dynamic, and often complicated, population structure over a considerable duration of time. Successfully accomplishing this involves addressing both the *temporal* and the *structural* characteristics of the data.

A *data model* is an abstract logical definition of the data that will be stored including a detailed description of its structure, and perhaps its behavior (Date, 2000). The data model provides the conceptual blueprint around which a working database system is built. Several existing data models successfully address the challenges posed by longitudinal population data, but with significantly different solutions, and as a result data stored using these different data models are hard to compare, merge or analyze together. Moreover when ongoing projects use different data models, it is difficult to design an investigation that utilizes or collects data from more than one project because their individual data models are largely incompatible. Multi-site longitudinal investigations are and will become more common in the future making it an urgent matter to develop a standard temporal framework to guide the storage and manipulation of complex temporal data describing the histories of people (and households and other aggregations of people) living in multiple populations.

This work begins to address this challenge by presenting 1) an abstract temporal data model that can represent an arbitrary range of inter-related temporal trajectories – the General Temporal Data Model or GTDM, 2) a relational database implementation of the GTDM that is able to store an arbitrary range of inter-related temporal trajectories with a single *static* relational schema – the Structured Population Event History Register or SPEHR, and 3) a relational database schema based on SPEHR that can store the contents of many SPHER-based databases allowing data from different longitudinal projects to be easily merged and managed together. The aim is threefold:

1. to improve the efficiency and effectiveness of data management at individual longitudinal projects,
2. to allow multiple longitudinal projects to easily share and compare data and implement multi-site investigations that make full use of both existing data and new data to be collected during the course of the new investigation, and
3. by providing a standardized data model, to allow the development of standardized and largely automated data manipulation, extraction and analysis tools.

1.1 Longitudinal Data Sources

Longitudinal data are generated through many different study designs, including:

- linked, repeated cross-sectional surveys,
- panel studies,
- revolving panel studies,
- cohort studies,
- population laboratories (community-level population surveillance),
- vaccine, intervention (prevention) and drug trials,
- environmental monitoring, and
- ecology laboratories, to name a few.

Each of these have in common that they make repeated observations of the units of analysis over time and link the data collected at each observation for each unit of analysis. Beyond this there is great variability, mainly in how the units of analysis are chosen, when and how they enter and exit the study, the detail of information collected, whether any information is collected that describes the interaction of individual units of analysis with each other, and finally, whether data are collected at more than one level of aggregation, perhaps at the household and community levels in addition to the individual level.

Among the most prominent long-running longitudinal population studies are the various DSS sites that have addressed a range of social science and health topics over the past century (See for example: Axinn et al., 1997; Binka et al., 1999; Binka et al., 1996; Clark et al., 1995; Clark, 2001a; Cliggett, 1997; Colson, 1960, 1964, 1971; Colson and Scudder, 1987; Desgrees du Lou et al., 1995; Forster and Snow, 1995; Forster, 1995; Garenne, 1995; Garenne et al., 1992; Garenne and Cantrelle, 1998; Garenne et al., 1997; Gray et al., 1997; Gray et al., 1998; Lamb et al., 1984; Linder, 1971; MacLeod et al., 1996; Pison et al., 1997; Poulsen et al., 1997; Scudder, 1962, 1985; Scudder and Colson, 1977, 1980; Shamebo et al., 1991; Shamebo et al., 1993; Tollman et al., 1999; Tollman and Zwi, 2000; Wyon and Gordon, 1971). Perhaps the most well known of these is the Matlab project in Bangladesh, initiated in the 1960s and collecting data continuously to the present time (ICDDR-B, 2004). Over the past decade a relatively large number of new DSS sites have been initiated to study a wide range of health and social issues. The INDEPTH network (INDEPTH Network, 1998, 2004b) was created in 1998 to bring the DSS sites together and provide a means through which they can share methods and results. INDEPTH now lists 34 DSS sites as full members with a small number of others as associate members (INDEPTH Network, 2004a). In spite of the substantial costs involved, the majority of those 34 sites have been initiated within the past five to ten years indicating a strong and growing interest in this methodology.

1.2 The Structure of Temporal Data

Temporal data have three main dimensions: 1) the unit of analysis, 2) the attributes (both constant and time-varying) of the unit of analysis, and 3) time. The *unit of analysis* dimension relates to the set of items being described by the data. The *attributes* dimension relates to the set of attributes that is attached to and describes each unit of analysis; some of which are constant with time while others vary. The *time* dimension relates to the temporal qualities of the data; that units of analysis come into existence, go out of existence, come under observation, leave observation, that some of their attributes change over time, and that their relationships to one another change over time. The time-evolving relationships between units of analysis describe the *structure* and dynamics of the population of units of analysis and are particularly difficult to model and manage in a general and efficient way.

In contrast, non-temporal data have only the first two dimensions: 1) the unit of analysis, and 2) the attributes. These can be easily represented with the traditional two-dimensional table; each column of the table associated with an attribute, and each row of the table representing a different unit of analysis. Each column has a domain of values corresponding to the possible values of the attribute, and each row is an assertion that combines a valid value from each attribute domain to describe a unique unit of analysis.

Adding the *time* dimension effectively adds a third dimension to the table so that the value of each attribute for each unit of analysis is recorded at all times; one can visualize this as a stack of two-dimensional tables with one two-dimensional layer for each instant in time. As conceptually elegant as this may seem, it is totally impractical for many reasons. Time is smooth and unbounded which would require an infinite number of layers in this stack, and many of the attribute values would be repeated many times for those attributes that remain constant over time. This would require an infinite amount of storage and would result in vast replication of data with the potential for inconsistencies to arise. Moreover, this model does not provide a tractable means through which to represent collections of the units of analysis, their relationships to one another as they change through time, or their relationships to/with units of analysis of a different type. For example, people's memberships in households and villages, and their residencies at different locations.

Clearly, traditional two-dimensional tables are inadequate, and a naïve sequencing of two-dimensional tables leads immediately to significant limitations in the ability to represent interesting facts, opens up the potential for significant corruption of the data through inconsistent representations of the same fact, and finally requires a vast, redundant excess of storage capacity.

1.3 Existing Temporal Frameworks

Although much effort has been expended over the past two decades developing various conceptual frameworks for temporal data (See for example: Allen, 1983; Allen and Ferguson, 1994; Date et al., 2002c; Etzion et al., 1998; International Organization for Standardization, 2000; Jensen, 2000; Jensen et al., 1998; Snodgrass, 2000; Snodgrass et al., 1998; Spaccapietra et al., 1998), almost none of this work has been incorporated into working database systems that are widely available (A promising implementation of the temporal extensions to the Relational Model suggested by Date et al. is underway by Alphora, 2004). In spite of this lack of a standard temporal database model and methodology, this work has yielded widely useful results and seems on the verge of coalescing into a standard temporal database framework. A standard terminology has been identified and is available as the "Consenus Glossary of Temporal Database Concepts" (Jensen et al., 1998), a basic set of standard temporal primitives and operators have been defined and largely incorporated into the current international standard for the Structured Query Language (SQL) (Gulutzan and Pelzer, 1999), and a recent volume co-written by one of the relational database's most prominent advocates, C.J. Date, is devoted entirely to the extension of the Relational Model of Data to incorporate explicitly temporal data (Date et al., 2002c).

While computer scientists have been developing database methodologies to handle temporal data, population and health scientists have developed their own solutions to these challenges using existing database tools. The majority of these have been developed on an *ad hoc* basis in-house by individual groups who needed an immediate solution to a specific data management need, and consequently they are each unique and it is not possible to easily share and compare the data that they manage, or in most cases to even understand on what principles they operate because no published documentation is available.

A prominent exception is the Household Registration System (HRS) developed by Bruce MacLeod and colleagues in conjunction with the Navrongo DSS site in Ghana (MacLeod et al., 1996; Phillips et al., 2000). The HRS forms the basis of the data management system used by ten or so of the DSS sites who are members of the INDEPTH network, and as such, is the *de facto* standard data management system for DSS sites. Conceptually, the HRS is built around the Reference Data Model (RDM) (Benzler et al., 1998) which has been publicly available on the INDEPTH network's web site since 1998. The RDM is an explicitly temporal blueprint for a relational database that can record the history of a human population. It recognizes a number of key events that determine transitions in a human life and a number of key episodes that mark time intervals between events when a well-defined state is maintained. The RDM has the facility to record and manage social relationships, membership in social groups, residences at various locations, "status" observations, observation times, and all of the events necessary to define the population under observation and track its basic dynamics. A drawback of the RDM, and hence of the HRS, is that it is a largely rigid (inflexible) model that cannot be easily extended or modified without adding new components or substantially modifying existing ones. As a result, each site that uses the RDM (in the form of the HRS) has to invest time customizing the data model and the data management system to suit their own needs – albeit *much less* time and energy than if they developed their systems *de novo* instead of using the RDM and HRS as a starting point. The impact of this customization has been to create a number of different implementations of the RDM that are no longer compatible with each other; thus compromising one of the most important benefits to accrue from standardization – the ability to easily share and compare data stored in two or more systems based on the same standard.

Given the rapid growth in the number of intensive community-based longitudinal projects studying population and health issues, the concurrent wide dissemination of affordable computing capability, and the existing lack of easily-accessible, standardized temporal database tools; there is a strong and growing demand for a general, robust, easily-extensible temporal conceptual framework in which to organize and manipulate large quantities of temporal data. Moreover, a *standard* temporal model is necessary to unlock the potential synergy inherent in the ability to share and compare data collected by many sites, and the potential to create study designs that utilize more than one site.

2 Aim

This paper presents a general conceptual framework to unify the representation of time and the structure and description of complex collections of temporally interconnected "things". The fundamental components of a temporal system are identified, and both they and their relationships to each other are defined in a simple, standard way that is generalizable. A metadata framework is proposed to endow this abstract generalization with specific meaning and serves also to inseparably bind the definitions of the data to the data themselves.

The result is a temporal data model that is highly generalized, conceptually simple and complete, and inherently contains a full description of the primary data it manages. Consequently individual databases designed around the General Temporal Data Model (GTDM) can be fully customized to suit the needs of their owners without modifications to the underlying logical database schema or the physical implementation of the database in a database management system; while still offering the potential to transparently share and compare compatibly subsets of their data with other similar databases, and because the metadata fully describe the primary data stored in the system, the compatible (comparable) data in two or more databases can be readily identified.

The proposed GTDM has several additional benefits. The structure is highly normalized in the sense that facts are not stored more than once, thus preventing the potential for duplicate representations of facts to become inconsistent. In addition to documenting the primary data, the metadata also make it possible to create general operations that effect the primary data automatically, customizing their effects based on the information contained in the metadata. This provides the potential to automate many routine database management tasks. The GTDM naturally facilitates the definition of hierarchical groupings and is easily able to track the dynamics of these groups and their members. This is particularly useful when tracking the social dynamics of human populations (people's membership in households for example).

Although the GTDM is sufficiently general to be applied to any temporal system, we are particularly interested in its application to human populations in the context of population and health studies. The remainder of this paper will present:

- concepts of time and temporal relationships,
- an explicitly temporal model of reality,
- the Structured population event history register,
- a multi-site extension of the structured population event history register

Everything that follows is discussed in the context of population and health research conducted on human beings, and the methods presented are optimized for the collection, storage, and manipulation of data describing the history of human populations. This work has arisen out of the author's close collaboration with a number of DSS sites, and consequently all of the specific examples reflect to the DSS methodology.

3 Conceptualizing Time and Temporal Relationships

A longitudinal enterprise is "longitudinal" because it "extends over a period of time" (Merriam-Webster, 1996), making the measurement and manipulation of time a fundamental component of such an enterprise, and consequently we begin by defining time and its measures. Refining these concepts with an understanding of how time is experienced in the real world leads naturally to an intuitive abstraction of temporal entities and relationships. The resulting abstraction is very general and can be implemented in a number of ways using a wide range of technologies.

3.1 Time, Measures of Time, and Valid Time[1]

Time is *universal, one-dimensional, dense* and *unbounded*. A single time domain exists at all locations, leading to a general notion of concurrence (*universal*). Individual elements constituting the time domain have no extent within the domain (zero duration) and are unambiguously identified by ordered, unique values of a single, numeric attribute called "position" (*one-dimensional*). Between any two elements it is possible to insert an additional element (*dense*), and given this, it is always possible to insert a new element *before* the first and *after* the last element (*unbounded*). The time domain is what is known colloquially as the *time line*. The foregoing paragraph is adapted from Benzler and Clark (Under Review 2004).

3.1.1 Measures of Time

There are five fundamental measures of time. These allow us to conceptualize and manipulate the time domain irrespective of the meaning that may be associated with time.

[1] Section 3.1 is adapted from another manuscript currently under review by Justus Benzler and the author: Benzler, J. and S. J. Clark. Under Review 2004. "Towards a Unified Timestamp with Explicit Precision." *Demographic Research*. Terminology in this section is largely consistent with that introduced in "Towards a Unified Timestamp with Explicit Precision" but in some cases deviates from that in "The Consensus Glossary of Temporal Database Concepts": Jensen, C. S., C. E. Dyreson, M. Bohlen, J. Clifford, R. Elmasri, S. K. Gadia, F. Grandi, P. Hayes, S. Jajodia, W. Kafer, N. Kline, N. Lorentzos, Y. Mitsopoulos, A. Montanara, D. Nonen, E. Peressi, B. Pernici, J. F. Roddick, N. L. Sarda, M. R. Scalas, A. Segev, R. T. Snodgrass, M. D. Soo, A. Tansel, P. Tiberio, and G. Wiederhold. 1998. "The Consensus Glossary of Temporal Database Concepts - February 1998 Version." in Temporal Databases: Research and Practice, edited by O. Etzion, S. Jajodia, and S. Sripada. Berlin: Springer.

3.1.1.1 Time Element

A time element is one of the basic elements constituting the time domain. A time element can be located at any position within the time domain and has zero duration.

3.1.1.2 Time Point

A time point identifies a single element in the time domain with a position. Because it simply identifies a time element, a time point also has zero duration. A time point is labeled with a single numeric value.

3.1.1.3 Time Duration

A time duration identifies an extent within the time domain. A time duration does not have a position within the time domain, nor is a time duration a single set of time elements. A time duration is labeled with a single numeric value denoting its extent in the time domain. Because the elements of the time domain are ordered, a time duration can have directionality; can extend in either direction along the time domain.

3.1.1.4 Time Interval

A time interval identifies both an extent and a position within the time domain. A time interval consists of a bounded, infinite set of time elements within the time domain. A time interval is labeled with either: 1) two single numeric values corresponding to the positions of its first and last time elements, or 2) a single numeric value corresponding to its first (or last) time element and a single numeric value corresponding to its duration (extent in the time domain).

3.1.1.5 Time Set

A time set identifies a finite set of non-overlapping time intervals within the time domain.

3.1.2 Measures of Time with *Meaning*: Valid Time

Here, *meaning* refers to the salience of a fact; something that is both perceived and important in the scope of human experience. Combining this notion of meaning with measures of time yields Valid Time (VT). More precisely, VT is the time when a fact is true in the real world (or a modeled reality)[2] (Jensen et al., 1998). VT associates a true proposition with a measure of time, and consequently, VT also takes five basic forms.

3.1.2.1 Instant

An instant is the association of a fact with a time element. The meaning of an instant is well defined while its position in the time domain is not. Like a time point, an instant has zero duration.

3.1.2.2 Event

An event is the association of a fact with a time point. Both the meaning of an event and its position in the time domain are well defined.

3.1.2.3 Period

A time period is the association of a fact with a time duration. The proposition that remains true throughout a period and the duration of the period are well defined, while the position of the period within the time domain is not.

[2] This notion of VT differentiates it from Transaction Time (TT) which corresponds to the time when VT is recorded in a specific database. For example the VT time interval during which someone was alive may have been June 1, 1971 to August 2, 1994. If this VT time interval was recorded in a specific database on February 2, 1995 and persists in the database until the present, the TT for that VT time interval is the open-ended time interval beginning February 2, 1995. For a discussion of VT, TT and how they might be captured and manipulated in a temporal database see: Date, C. J., H. Darwen, and N. A. Lorentzos. 2002c. Temporal Data and the Relational Model. San Francisco: Morgan Kaufmann, Snodgrass, R. T. 2000. Developing Time-Oriented Database Applications in SQL. San Francisco: Morgan Kaufmann Publishers.

3.1.2.4 State

A state is the association of a fact with a time interval. The proposition that remains true throughout the state, the duration of the state and the position of the state within the time domain are all well defined.

3.1.2.5 Pattern

A time pattern is the association of a fact with a time set. Both the fact that is true during the time intervals that constitute the pattern, and the extent and position of those time intervals within the time domain are all well defined.

3.2 Measures of Time in Practice and Precision

In practice we use both the raw measures of time and the measures of VT. For example, the raw measures of time are manipulated to construct various calendars and the systems used to convert between them, and are also used to construct and calibrate clocks.

However, it is the VT measures that are most common in our daily lives and most likely to be stored and manipulated in a database. We routinely refer to events that effect us; births, deaths, marriages, divorces, the start of the work day, or the end of the month, etc. Likewise, states are a natural part of our everyday vernacular; Jack and Jill's marriage, the life of Mozart, or the Second World War, etc. The other VT measures are perhaps less obvious, but they are also part of our daily lives. A state of having flu that you had last year *sometime* is a duration, although in this case the uncertainty about its position in the time domain is bounded. A woman's pregnancy states compose a pattern, and instants often describe marginally salient events whose position in the time domain is not well defined, such as the purchase of a lava lamp *sometime* in the past.

In reality, the measurement of time is a messy business and the theoretical precision assumed in the preceding sections is never possible. Instead every fact is mapped to the time domain with some degree of fuzziness. For example, we cannot pinpoint the precise, zero duration time element associated with a birth, but we can put bounds on when the birth took place. Likewise all events are recorded with some degree of precision that is never perfect but often knowable. In order to preserve and store the maximum amount of information and to exclude any implicit assumption about precision, a new type of timestamp is necessary that effectively stores all time points as intervals that correspond to the degree of imprecision associated with the measurement of the time point. A full discussion of temporal measures with explicit precision is presented elsewhere by Justus Benzler and the author (Benzler and Clark, Under Review 2004).

4 A General Temporal Model of Reality

Time is such a basic and invisible element of our lives that we take it for granted and rarely think deeply about it. This deceptive simplicity often leads modelers and database designers to include time as an afterthought or *ad hoc* component of their models, and this has led to tortuously complex and difficult to manage databases. It also often results in significant duplication of (temporal) information, opening up the potential for (temporal) data within the same database to become inconsistent and thereby corrupt. To begin rectifying this there is substantial ongoing work in the area of temporal databases, and standards governing the basic measures of time and their electronic representations are being proposed and evaluated as part of the current international standards for data manipulation languages[3]. These do not, however, propose standard ways of modeling temporal reality.

[3] Proposed additions to the most recent edition of the SQL standard (SQL-99) include standard definitions of temporal measures and some standard methods to manipulate them, see: Snodgrass, R. T., M. H. Bohlen, C. S. Jensen, and A. Steiner. 1998. "Transitioning Temporal Support in TSQL2 to SQL3." in Temporal Databases: Research and Practice, edited by O. Etzion, S. Jajodia, and S. Sripada. Berlin: Springer.

A temporal model is one that explicitly considers time as one of its components. The fundamental challenge emerges from the fact that time is universal, dense and unbounded, whereas almost all other entities in the model are *discrete* within both time and space and thereby only able to effect or be connected to a finite number of other entities – time effects everything, everything experiences time, time has no bounds and time can be resolved with infinite precision. The following sections develop the core concepts necessary to model a temporal reality, present an abstract conceptualization of the interaction between time and the discrete entities we wish to model, and finally propose a general, integrated temporal model of reality.

4.1 Temporal Entities: States

All entities that we wish to consider have a valid *lifetime*. This makes them states in the sense described above in 3.1.2.4. They all have a well defined beginning (start) and end (stop) with a *constant meaningful* state between those; most generally the state of “existing”. Temporal entities represent some*thing* of interest and are associated with a specific temporal construct, a state; and consequently we generalize the term *state* and use it as a concise name for temporal entities. *States* can represent both physical and nonphysical entities. Tangible entities of common interest include people, places, and physical items like houses and cars; while often important nonphysical entities include marital unions between people, household unions between groups of people and the period of residence at a location.

The common characteristics of all states are: 1) a constant, meaningful “mode or condition of being” (Merriam-Webster, 1996), and 2) the association of that state with a known duration and position within the time domain.

4.2 Temporal Junctures: Events

Events bring about the temporal change that we wish to consider. As defined above in 3.1.2.2 an event is a meaningful happening associated with a well defined time point. Implicit in the meaning of an event is the change that it represents, and it is this notion of change that is most important to us. Events bring about or signify the beginning and ending of all states. Other events influence existing states without beginning or ending them, but these too can be understood as the beginnings and endings of other states that are in turn sub-states of the parent state influenced by the event.

Perhaps more important but less obvious is the fact that events form and dissolve all of the relationships that exist between states, and in this way they provide the means through which all states are joined – hence their description as temporal junctures – “1: joint, connection; 2: an instance of joining : union, junction; 3: a point of time; esp: one made critical by a concurrence of circumstances” – (Merriam-Webster, 1996). For example a birth effects, potentially: the infant, the mother, the father, the place where the birth takes place, and the existing siblings. In this example, all of these states are now linked to each other as a result of the occurrence of the birth; some were previously linked in different ways through the occurrence of a wedding and other births, but this new birth changes something for all of them and sets up a new set of links. Likewise, it signals the end of various sub-states for the different states, such as the sub-state of having n-1 siblings for each of the existing siblings.

The common characteristics of all events are: 1) something changes, 2) the association of that change with a specific time point in the time domain, and 3) the fact that this change influences at least one (and likely several) states. In the same way that states describe all that is *constant*, events describe all *changes*, and in addition provide the means through which to *join* states in a temporally consistent fashion.

4.3 Temporal Nexus: Influences

Influences are the explicit representation of the links between states and the events that *influence* them. Because it is possible for an event to influence more than one state, each event can be linked to many states, and through their individual links to the event, those states are all linked to each other.

In this way influences form a temporal nexus – “1: connection, link; 2: a connected group or series; 3: center, focus” (Merriam-Webster, 1996) – between states and the events that influence them. The individual event (life) histories of the states intersect when an event influences two or more states, and these intersections represent the formation and dissolution of relationships between the states.

The common characteristics of all influences are: 1) they represent the *influence* – 1: to affect or alter by indirect or intangible means : sway; 2: to have an effect on the condition or development of: modify (Merriam-Webster, 1996) – of a specific type of event on a specific type of state, and 2) they must be linked to exactly one event and exactly one state.

4.4 An Example: Event ⬄ Influence ⬄ State

To illustrate, imagine we are interested in recording the vital events, nuptial histories and migratory behavior of a small number of people living in New York and London. We initiate a small study at time 0.5 and enroll the two locations, New York and London; two people, Richard and Elizabeth; and because Richard and Elizabeth live in London, we initiate a residency for each at London. At time 3.0 a wedding occurs in London that joins Richard and Elizabeth as a couple and initiates their marital union. At time 6.5 in London Elizabeth gives birth to Beatrice, and at time 9.0 we visit our “sites” and make an observation of all the existing entities enrolled in our study. At time 12.5 Richard dies in London, and as a result of Richard’s death, Elizabeth and Beatrice move from London to New York at time 15.5. At time 17.8 we visit our “sites” again and make another observation of all the existing entities enrolled in our study, and the study continues to the present time, 19.0. Throughout the study we organize and record the information as events, influences and states.

Figure 1: Diagram of Event ⬄ Influence ⬄ State Example

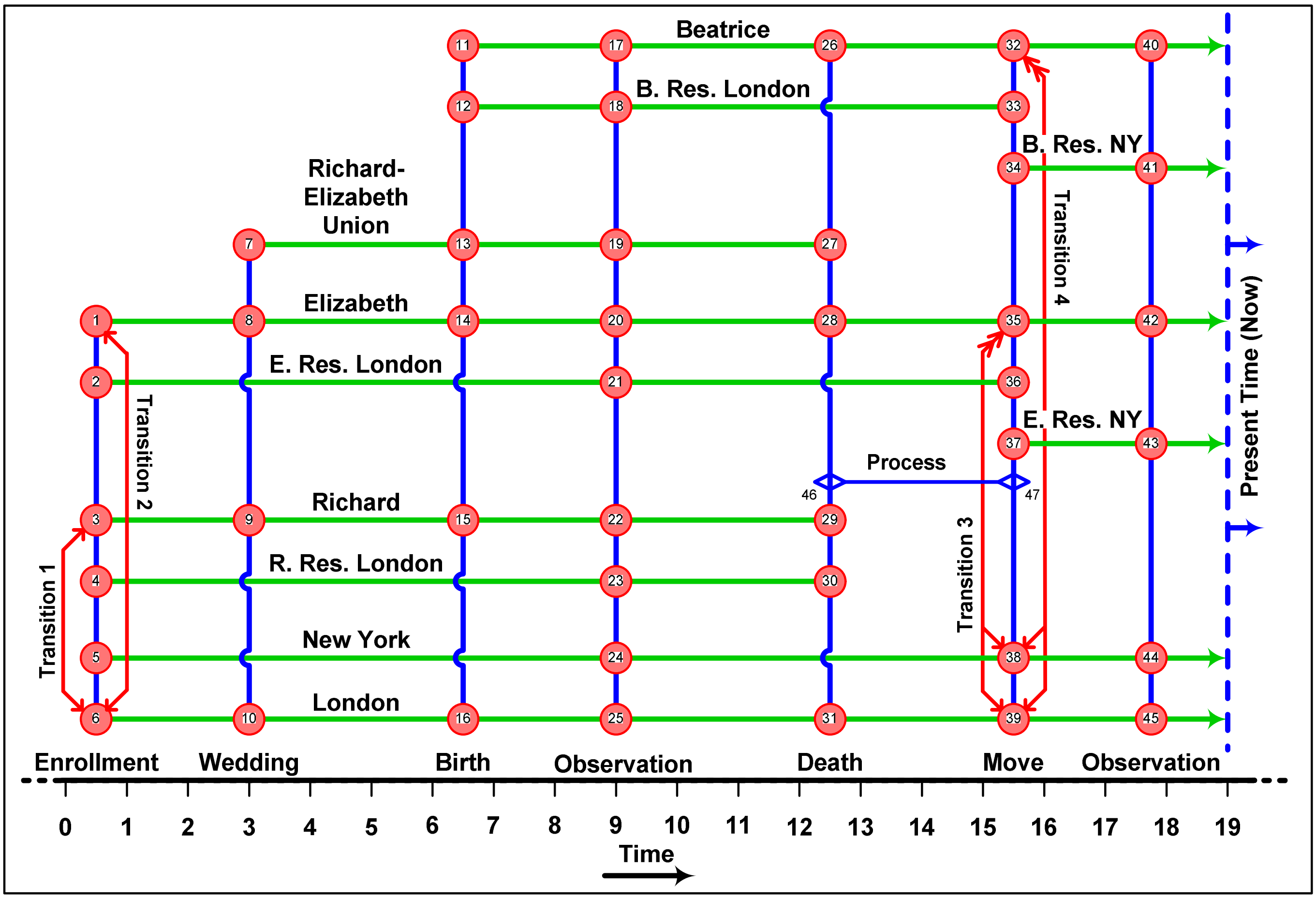


Table 1: Influences in London-New York Event ⬄ Experience ⬄ State Example

	Events													
States		Enrollment		Wedding		Birth		Observation @ 9		Death		Move		Observation @ 17
Beatrice					11	a. Person start: birth b. Person, residence start: birth	17	a. Person, observed: observation	26	a. Person, parent dies: death	32	a. Person, move away from: move (London) b. Person, move to: move (New York) c. Person, residence stop: move (London) d. Person, residence start: move (New York)	40	a. Person, observed: observation
Beatrice Residence at London					12	a. Residence start: birth	18	a. Residence, observed: observation			33	a. Residence stop: move		
Beatrice Residence at New York											34	a. Residence start: move	41	a. Residence, observed: observation
Richard-Elizabeth Union			7	a. Union start: wedding	13	a. Union, child born: birth	19	a. Union, observed: observation	27	a. Union stop, spouse dies: death				
Elizabeth	1	a. Person, enroll: enrollment b. Person, residence start: enrollment (London) c. Person, at place: enrollment (London)	8	a. Person, marry: wedding	14	a. Person, child born: birth	20	a. Person, observed: observation	28	a. Person, spouse dies: death	35	a. Person, move away from: move (London) b. Person, move to: move (New York) c. Person, residence stop: move (London) d. Person, residence start: move (New York)	42	a. Person, observed: observation
Elizabeth Residence at London	2	a. Residence start: enrollment					21	a. Residence, observed: observation			36	a. Residence stop: move		
Elizabeth Residence at New York											37	a. Residence start: move	43	a. Residence, observed: observation
Richard	3	a. Person, enroll: enrollment b. Person, residence	9	a. Person, Marry: wedding	15		22	a. Person, observed: observation	29	a. Person stop: death b. Person,				

	Events													
States	Enrollment		Wedding		Birth		Observation @ 9		Death		Move		Observation @ 17	
		start: enrollment (London) c. Person, at place: enrollment (London)								residence stop: death				
Richard Residence at London	4	a. Residence start: enrollment					23	a. Residence, observed: observation	30	a. Residence stop: death				
New York	5	a. Place, enroll: enrollment					24	a. Place, observed: observation			38	a. Place, person moves to: move (Beatrice) b. Place, person moves to: move (Elizabeth) c. Place residence start: move (Beatrice) d. Place, residence start: move (Elizabeth)	44	a. Place, observed: observation
London	6	a. Place, enroll: enrollment b. Place, residence start: enrollment (Elizabeth) c. Place, residence start: enrollment (Richard) d. Place, person at: enrollment (Elizabeth) e. Place, person at: enrollment (Richard)	10	a. Place, wedding: wedding	16	a. Place, child born: birth b. Place, residence start: birth	25	a. Place, observed: observation	31	a. Place, person dies: death b. Place, residence stop: death	39	a. Place, person moves from: move (Beatrice) b. Place, person moves from: move (Elizabeth) c. Place residence stop: move (Beatrice) d. Place, residence stop: move (Elizabeth)	45	a. Place, observed: observation
Process: Richard Death → Elizabeth & Beatrice Move									46	a. Process death-move start: death	47	a. Process death-move stop: move		

Figure 1 displays a diagram of the information we collect. Time is recorded on the horizontal axis and is marked with equidistant positional markers from 0-19. Horizontal (green) lines represent states, vertical (blue) lines represent events and the shaded (red) circles at the intersection of horizontal and vertical lines represent (potentially multiple) influences. States and events are labeled with descriptive labels while intersections between states and events where influences occur are numbered. Descriptions of the influences at each numbered intersection of a state and an event are contained in Table 1. Sometimes there is no relevant connection between a state and an event, and in that case the lack of an influence is indicated with a "jump" (semi-circular intersection symbol) instead of a shaded circle. The influence descriptions are written with reference to the states indicating how each state *is influenced* by each event. The horizontal line with diamond-shaped ends that connects the death and migration events indicates that we know that they are causally linked; that they are part of a *process* that occurs over a defined period of time. The vertical lines tangent to and to the right and left of influences associated with the initial enrollment and the move events indicate that the connected influences are part of a *transition*; belong to a collection of related influences that describe movement out of and into *collections* – more on these terms in section 5 below.

4.5 The General Temporal Data Model

In its most abstract form the General Temporal Data Model is simply the **Event** ⇔ **Influence** ⇔ **State** triad that allows one to associate states with events that influence them and with other states with which they share some sort of relationship.

A *state* is an abstract generalization of any well-defined state that persists for a defined duration of time, including both physical things (such as a person) and non-physical states (such as a marital union that exists between two people). An *event* is any well-defined state-altering occurrence at a specific time. All transitions between states are marked in time and given meaning by the event with which they are associated, and consequently, a complete definition of an event includes a precise description of *all* its influences on various previously defined states. Linking the states to the events that start, effect and stop them are the *influences*. An influence links a state to an event that influences it and records the nature of the influence, or the *meaning* of the linkage.

Because all connections between things are initiated and broken through the occurrence events, and because it is possible to associate an arbitrary number of states with each event and an arbitrary number of events with each state (a *many-to-many* link between states and events), it is possible via their connections to the same events to record all of the connections between states. Moreover, since the connections between states must include an event, the time-dependence of the processes that form and break the connections is inextricably part of the connections themselves. This simple three-part abstraction elegantly represents both the *structure* and the *dynamics* (temporal aspects) of a time-evolving collection of states – an *arbitrary* population.

The GTDM is capable of storing the history and time-evolving structure of an arbitrary collection of states at whatever level of detail required (more detail requires the definition of more states, events and influences). Additionally the structure of the GTDM naturally facilitates the generation of *event lists* describing the history of any type of state, or indeed any individual state within a population. These lists of events are the basis of many types of longitudinal, survival or event history analysis and are easy to generate in a GTDM framework.

There are many different ways to realize this abstraction and implement it in a working system, most of the variation having to do with how attributes of different types of states and events are conceptualized, stored and manipulated. Following is a detailed description of a suggested complete realization of the GTDM intended for implementation in a relational database.

5 The Structured Population Event History Register

The Structured Population Event History Register (SPEHR) is a relational database schema based on the GTDM. Although the GTDM is fully general and can record the related histories of any type of "thing", SPEHR is a GTDM realization adapted to record the related histories of human beings, their residences at various locations and their memberships in various social groups. SPEHR retains the inherent schema-invariant flexibility and conceptual integrity of the GTDM but adds several features that are necessary in a working realization of the GTDM. As a logical blueprint for a relational database SPEHR can be implemented using *any* standard relational database management system[4] (Postgress, MySQL, MS SQL Server, IBM DB2, Oracle, MS Access etc.), and the object of this work is to present the relational schema for SPEHR (the blueprint) rather than provide details on implementing SPEHR in a specific relational database management system.

Figure 2 contains an entity relationship diagram of SPEHR that the reader may find useful while reading the following sections. The text that follows provides a conceptual overview of the SPEHR schema along with enough specific definition to launch someone on developing a SPEHR-based database. Detailed discussion of nitty-gritty design issues such as the precise construction of keys, indexes and referential integrity rules is omitted and left to the individual designer. However, Figure 2 contains primary key, foreign key and unique index definitions used by the author in the working examples of SPEHR implemented in MS Access that accompany this work. These examples provide the author's first pass at resolving these issues to make SPEHR actually work; albeit with a pedantic rather than operational focus.

5.1 Metadata

Metadata are "data about data" or data that describe other data. The concept of metadata is applied throughout SPEHR and is the critical element that allows SPEHR to realize the generality of the GTDM. The GTDM specifies three general objects – *States*, *Events* and *Influences* – that can each have many different *specific* types depending on the reality being represented by a GTDM-based database. SPEHR uses metadata to specify the domain of possible types for each of these general objects, and furthermore to specify the unique specific type of each instance of the general objects.

The practical result is that SPEHR contains a number of tables that contain metadata. Each row in a metadata table describes one unique type of a general object, and each metadata table is associated with a table whose rows contain instances of the general object. Each row in the general object table is a unique instance of the general object and must be associated with a specific type of that general object. This is accomplished by linking each row in the general object table to exactly one row in the associated metadata table, thereby identifying the specific type of each instance of the general object stored in the general object table.

By defining the appropriate referential integrity rule between the general object and metadata tables it is possible to insure that each instance of a general object (row in the general object table) is associated with exactly one type of that general object (row in the metadata table).

In the entity relationship diagram in Figure 2 metadata tables have unshaded headings with names in *italic* letters. The names of metadata tables are for the most part suffixed with "_Types" to indicate that the table in question contains a list of the specific types that define the domain of a general object.

5.2 States

SPEHR uses two tables to realize the *states* defined by the GTDM, enclosed in a box labeled "States" in Figure 2.

5.2.1 Table: State_Types

The metadata table named ***State_Types*** contains one row for each specific type of state that an individual SPEHR database can understand and record. Attributes of the ***State_Types*** table include:

[4] The only requirement of the database management system is that it be truly *relational* and support standard SQL. There are a wide variety of commercial and open source relational database management systems available; all have strengths and weaknesses, and it is for the individual organization to decide which is best suited to their enterprise.

- a unique identifier for each state type: **STID**,
- a name for each state type: **State**, and
- a brief description of each state type: **Description**.

5.2.2 Table: States

The table named **States** corresponds to the general object *states* specified in the GTDM and contains one row for each instance of a state that is recorded in an individual SPEHR database. Attributes of the States table include:

- a unique identifier for each state: **SID**, and
- the unique identifier of the state type associated with each state: **STID**.

A *many-to-one* referential integrity relationship exists between the **STID** attribute of the **States** table and the **STID** attribute of the ***State_Types*** table insuring that each state stored in the **States** table is associated with a valid state type stored in the ***State_Types*** table.

The **States** table can contain many rows of the same type (for example, type ‘person’), but each of these must correspond to a unique state of that type in the real world, and to reflect that, each will have a unique value in the ID attribute.

5.3 Events

SPEHR uses two tables to realize the *events* defined by the GTDM, enclosed in a box labeled “Events” in Figure 2.

5.3.1 Table: Event_Types

The metadata table named ***Event_Types*** contains one row for each specific type of event that an individual SPEHR database can understand and record. Attributes of the ***Event_Types*** table include:

- a unique identifier for each event type: **ETID**,
- a name for each event type: **Event**, and
- a brief description of each event type: **Description**.

5.3.2 Table: Events

The table named **Events** corresponds to the general object *events* specified in the GTDM and contains one row for each instance of an event that is recorded in an individual SPEHR database. Attributes of the **Events** table include:

- a unique identifier for each event: **EID**,
- the unique identifier of the observation event that recorded each event: **OEID**,
- the unique identifier of the event type associated with each event: **ETID**, and
- a timestamp that locates the event within the time domain (time line).

A *many-to-one* referential integrity relationship exists between the **ETID** attribute of the **Events** table and the **ETID** attribute of the ***Event_Types*** table insuring that each event stored in the **Events** table is associated with a valid event type stored in the ***Event_Types*** table.

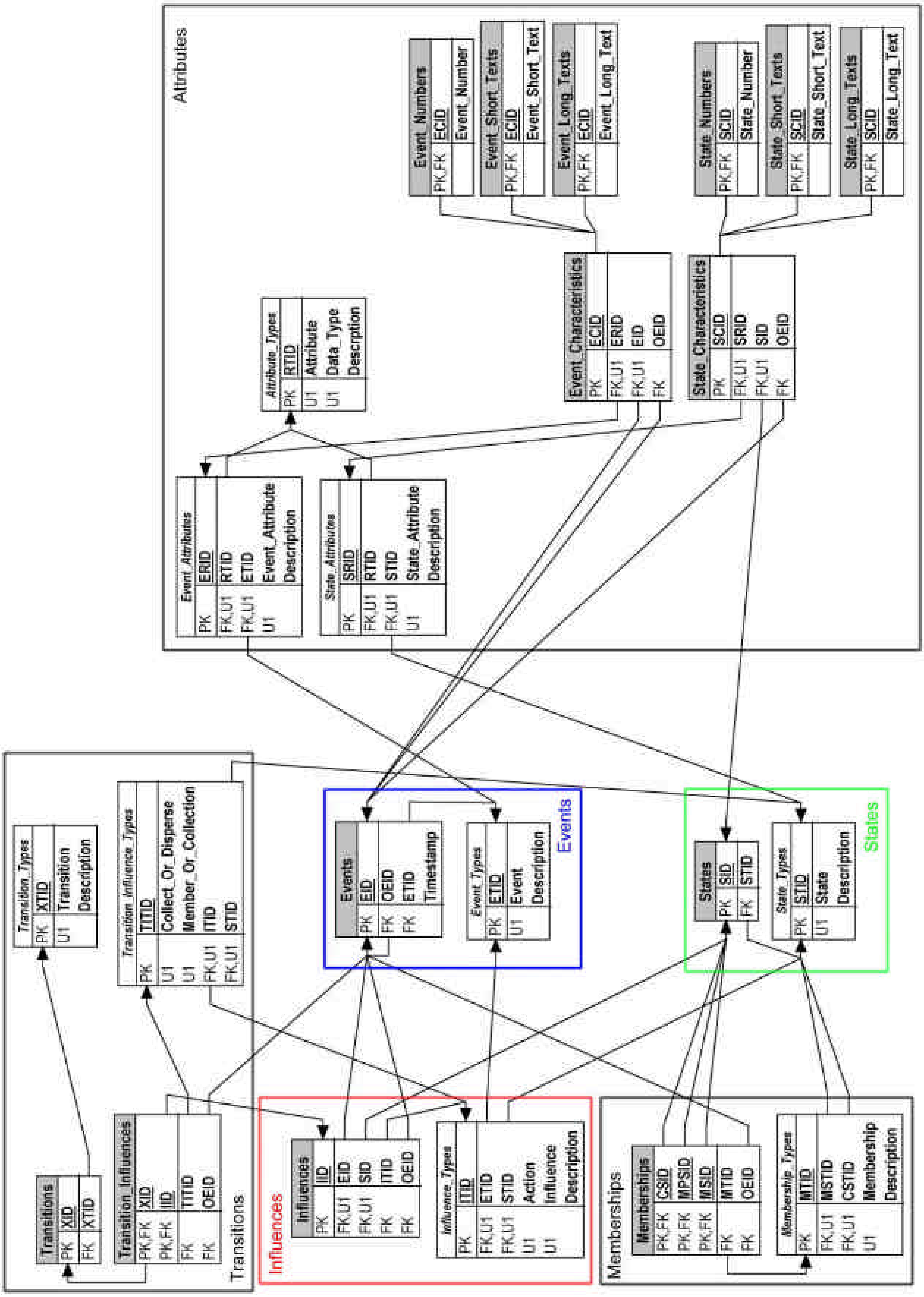
Likewise, a *many-to-one* (self) referential integrity relationship exists between the **OEID** and **EID** attributes of the **Events** table insuring that each event stored in the **Events** table is associated with the observation event at which it was recorded (observation events are linked to themselves).

The **Events** table can contain many rows of the same type (for example, type ‘birth’), but each of these must correspond to a unique event of that type in the real world, and to reflect that each will have a unique value in the ID attribute.

5.4 Influences

SPEHR uses two tables to realize the *influences* defined by the GTDM, enclosed in a box labeled “Influences” in Figure 2.

Figure 2: Entity Relationship Diagram of SPEHR



5.4.1 Table: Influence_Types

The metadata table named ***Influence_Types*** contains one row for each specific type of influence that an individual SPEHR database can understand and record. Attributes of the ***Influence_Types*** table include:

- a unique identifier for each influence type: **ITID**,
- the unique identifier of the event type associated with each influence type: **ETID**,
- the unique identifier of the state type associated with each influence type: **STID**,
- the *action type* associated with each influence (*start*, *stop* or *during* a state): **Action**,
- a name for each influence type: **Influence**, and
- a brief description of each influence type: **Description**.

A *many-to-one* referential integrity relationship exists between the **ETID** attribute of the ***Influence_Types*** table and the **ETID** attribute of the ***Event_Types*** table insuring that each influence type stored in the ***Influence_Types*** table is associated with a valid event type stored in the ***Event_Types*** table. Correspondingly a *many-to-one* referential integrity relationship exists between the **STID** attribute of the ***Influence_Types*** table and the **STID** attribute of the ***State_Types*** table insuring that each influence type in the ***Influence_Types*** table is associated with a valid state type stored in the ***State_Types*** table.

5.4.2 Table: Influences

The table named **Influences** corresponds to the general object *influences* specified in the GTDM and contains one row for each instance of an influence that is recorded in an individual SPEHR database. Attributes of the **Influences** table include:

- a unique identifier for each influence: **IID**,
- the unique identifier of the state associated with each influence: **SID**,
- the unique identifier of the event associated with each influence: **EID**,
- the unique identifier of the influence type associated with each influence: **ITID**, and
- the unique identifier of the observation event that recorded each influence: **OEID**.

A *many-to-one* referential integrity relationship exists between the **ITID** attribute of the **Influences** table and the **ITID** attribute of the ***Influence_Types*** table insuring that each influence stored in the **Influences** table is associated with a valid influence type stored in the ***Influence_Types*** table.

Additionally, a *many-to-one* referential integrity relationship exists between the **EID** attribute in the **Influences** table and the **EID** attribute in the **Events** table, and a *many-to-one* referential integrity relationship exists between the **SID** attribute of the **Influences** table and the **SID** attribute of the **States** table. These insure that each influence stored in the **Influences** table is associated with one valid event stored in the **Events** table and one valid state stored in the **States** table.

Last, a *many-to-one* referential integrity relationship exists between the **OEID** attribute in the **Influences** table and the **EID** attribute of the **Events** table insuring that each influence stored in the **Influences** table is associated with the observation event at which it was recorded.

The **Influences** table can contain many rows of the same type (for example, type ‘Person start: birth’), but each of these must correspond to a unique influence of that type in the real world, and to reflect that each will have a unique value in the ID attribute.

The metadata contained in the ***Influence_Types*** table are the heart of SPEHR. They contain most of the critical information that define what kinds of relationships a SPEHR database can represent. As a result defining the metadata in the ***Influence_Types*** table is potentially difficult and must be thought through carefully.

5.5 Static Attributes

At this stage the reader may be asking “where are the attributes” – things like “name”, “sex”, “birth weight”, “monthly expense” etc. Because the three primary data tables presented so far – **States, Events** and **Influences** – can contain many different *types* of the general objects they store, they do not have attributes of their own to contain idiosyncratic descriptive data of this sort, whose type and definition can and does change depending on the type of object instantiated in each row. For example, a row storing a state of type *person* may have additional attributes to describe the person's name, sex and place of birth; whereas a row storing a state of type *marital union* may not require any additional attributes, and a row storing a state of type *place* may require only one additional attribute in which to store the place's name. It is reasonable to assume that both events and states will need additional type-specific attributes, while influences will not because their type fully defines them.

Consequently the core SPEHR schema as defined above by the **States, Events** and **Influences** tables must be augmented with the facility to store arbitrary additional attributes for events and states. Above it has been reiterated several times that the fundamental point of developing SPEHR is to create an inherently temporal database schema that can model the arbitrary inter-related *histories* of a population of human beings, or more generally any population of "things" that may or may not interact with each other over time. The SPEHR that has been developed thus far is inherently temporal giving us the ability to fully track the history and dynamics of individual "things", and arbitrary collections of things (see 5.6 below). As a result any time-varying attribute of any "thing" must be modeled as a collection of states that are associated with the thing and *not* as an attribute of a state. By definition the *additional attributes of states are non time*-varying or **static** attributes that remain valid throughout the duration of the state. The additional attributes of events further describe the events, which are by definition non time-varying entities because they occur *at* a specific time point (see 3.1.2.2 above).

The additional attributes of events and states are stored in a collection of separate tables and linked to the specific events and states that they describe through a set of linking tables. Three metadata tables contain descriptions of the types of attributes that can be understood by an individual SPEHR database. A general metadata table named ***Attribute_Types*** contains the domain of all possible attributes, including their names and data types ("storage" types used by the individual database management system). Many of these can be applied to both events and states so two more metadata tables named ***Event_Attributes*** and ***State_Attributes*** precisely define the domains of attributes that can be applied to events and states. Each of these takes a fundamental attribute type from the metadata table ***Attribute_Types***.

The attribute values themselves are stored in two collections of tables, one for events and one for states, with (potentially) one table for each storage type supported by the database management system. In practice the database designer will decide which subset of the supported storage types are necessary for the SPEHR database being designed. The attribute values stored in these tables are linked to the events and states themselves through two more tables, one for events and one for states, that link individual events and states to their individual attribute values and take from the metadata tables, ***Event_Attributes*** and ***State_Attributes,*** the type of attribute that is being linked.

Many advantages and a one significant disadvantage accrue from this design of the attributes module of SPEHR. The advantages include the ability to handle an arbitrary and dynamic set of attributes for events and states, and the fact that the metadata effectively comprise a data dictionary that is inextricably associated with the primary data, so much so that if it were removed the primary data would become meaningless. The first advantage is significant because it allows the definition and *addition* of arbitrary numbers of attributes either at design time *or later without making modifications to the database schema,* and it allows legacy data to be retained and to retain their meaning without compromising the ability to store similar data with new definitions. The second advantage is also significant in that the primary data (the attribute values themselves) cannot and will not ever be separated from the metadata that provide them with their meaning – **the data dictionary for a SPEHR database is built-in and will never be lost**.

The disadvantage is unfortunately also significant; namely that this rather abstract and complex way of storing static attributes makes the retrieval of attributes more difficult and potentially less efficient. This is the only disadvantage to the SPEHR schema that the author viewed as potentially fatal, and so an experiment was conducted to test the efficiency of retrieving attributes from a SPEHR database storing a very large number of states (one million) and characteristics (four million). The experiment was conducted using the MS Access 2000 relational database management system implementing a simple version of the SPEHR schema with all relevant tables in the attributes module properly indexed. The result was to conclusively demonstrate that arbitrary sets of attributes can be retrieved for arbitrary single events and states almost instantaneously. Retrieving arbitrary sets of attributes for large numbers of events and states takes slightly longer – on order seconds or minutes – which is acceptable because large “reports” of the type that require large numbers of events and states to be attributed do not typically need to run with extreme speed.

Following is a detailed description of the tables and relationships that comprise the attributes module of SPEHR.

5.5.1 Table: Attribute_Types

The metadata table named ***Attribute_Types*** contains one row for each specific (general) type of static attribute that an individual SPEHR database can understand and record. Attributes of the ***Attribute_Types*** table include:

- a unique identifier for each static attribute type: **RTID**,
- a name for each static attribute type: **Attribute**,
- a data type (“storage” type) for each static attribute type: **Data_Type**, and
- a brief description of each static attribute type: **Description**.

5.5.2 Table: Event_Attributes

The metadata table named ***Event_Attributes*** contains one row for each specific type of static *event* attribute that an individual SPEHR database can understand and record. Attributes of the ***Event_Attributes*** table include:

- a unique identifier for each static event attribute type: **SRID**,
- the unique identifier of the (general) attribute type associated with each static event attribute type: **RTID**,
- the unique identifier of the Event type associated with each static event attribute type: **STID**,
- a name for each static event attribute type: **Event_Attribute**, and
- a brief description of each static event attribute type: **Description**.

A *many-to-one* referential integrity relationship exists between the **RTID** attribute of the ***Event_Attributes*** table and the **RTID** attribute of the ***Attribute_Types*** table insuring that each event attribute stored in the ***Event_Attributes*** table is associated with a valid (general) attribute type stored in the ***Attribute_Types*** table.

Additionally, a *many-to-one* referential integrity relationship exists between the **ETID** attribute of the ***Event_Attributes*** table and the **ETID** attribute of the ***Event_Types*** table insuring that each event attribute stored in the ***Event_Attributes*** table is associated with a valid event type stored in the ***Event_Types*** table.

5.5.3 Table: State_Attributes

The metadata table named ***State_Attributes*** contains one row for each specific type of static *state* attribute that an individual SPEHR database can understand and record. Attributes of the ***State_Attributes*** table include:

- a unique identifier for each static state attribute type: **SRID**,
- the unique identifier of the (general) attribute type associated with each static state attribute type: **RTID**,
- the unique identifier of the state type associated with each static state attribute type: **STID**,
- a name for each static state attribute type: **State_Attribute**, and
- a brief description of each static state attribute type: **Description**.

A *many-to-one* referential integrity relationship exists between the **RTID** attribute of the ***State_Attributes*** table and the **RTID** attribute of the ***Attribute_Types*** table insuring that each state attribute stored in the ***State_Attributes*** table is associated with a valid (general) attribute type stored in the ***Attribute_Types*** table.

Additionally, a *many-to-one* referential integrity relationship exists between the **STID** attribute of the

State_Attributes table and the **STID** attribute of the ***State_Types*** table insuring that each state attribute stored in the ***State_Attributes*** table is associated with a valid state type stored in the ***State_Types*** table.

5.5.4 Table: Event_Characteristics

The table named **Event_Characteristics** records the links (relationships) between individual events and individual static event attribute values and contains one row for each such *event characteristic* that is recorded in an individual SPEHR database. Attributes of the **Event_Characteristics** table include:

- a unique identifier for each event characteristic: **ECID**,
- the unique identifier of the event attribute associated with each event characteristic: **ERID**,
- the unique identifier of the event associated with each event characteristic: **EID**, and
- the unique identifier of the observation event that recorded each event characteristic: **OEID**.

A *many-to-one* referential integrity relationship exists between the **ERID** attribute of the **Event_Characteristics** table and the **ERID** attribute of the **Event_Attributes** table insuring that each event characteristic stored in the **Event_Characteristics** table is associated with a valid event attribute stored in the **Event_Attributes** table.

Additionally, a *many-to-one* referential integrity relationship exists between the **EID** attribute of the **Event_Characteristics** table and the **EID** attribute of the **Events** table insuring that each event characteristic stored in the **Event_Characteristics** table is associated with a valid event stored in the **Events** table.

Last, a *many-to-one* referential integrity relationship exists between the **OEID** attribute of the **Event_Characteristics** table and the **EID** attribute of the **Events** table insuring that each event characteristic stored in the **Event_Characteristics** table is associated with the observation event at which it was recorded.

5.5.5 Event Data Tables: Event_Numbers, Event_Short_Texts, Event_Long_Texts

The SPEHR schema has from one to *n* tables to actually store the static attribute values linked to events, one for each native storage type supported by the database management system that is necessary to store static attributes of events. They all take the following fundamental form; examples are displayed in Figure 2 and implemented in the example databases that accompany this work.

The tables named **Event_<native storage type name>** contain one row for each specific event-associated static attribute value stored in a SPEHR database. Attributes of the **Event_<native storage type name>** tables include:

- the unique identifier of the event characteristic with which each static event attribute value is associated: **ECID**,
- a value for each static event attribute: **Event_<native storage type name>**,

One-to-one referential integrity relationships exists between the **ECID** attributes of the **Event_<native storage type name>** tables and the **ECID** attribute of the **Event_Characteristics** table insuring that each static event attribute value stored in the **Event_<native storage type name>** tables is associated with a valid event characteristic stored in the **Event_Characteristics** table.

5.5.6 Table: State_Characteristics

The table named **State_Characteristics** records the links (relationships) between individual states and individual static state attribute values and contains one row for each such *state characteristic* that is recorded in an individual SPEHR database. Attributes of the **State_Characteristics** table include:

- a unique identifier for each state characteristic: **SCID**,
- the unique identifier of the state attribute associated with each state characteristic: **SRID**,
- the unique identifier of the state associated with each state characteristic: **SID**, and

- the unique identifier of the observation event that recorded each state characteristic: **OEID**.

A *many-to-one* referential integrity relationship exists between the **SRID** attribute of the **State_Characteristics** table and the **SRID** attribute of the **State_Attributes** table insuring that each state characteristic stored in the **State_Characteristics** table is associated with a valid state attribute stored in the **State_Attributes** table.

Additionally, a *many-to-one* referential integrity relationship exists between the **SID** attribute of the **State_Characteristics** table and the **SID** attribute of the **States** table insuring that each state characteristic stored in the **State_Characteristics** table is associated with a valid state stored in the **States** table.

Last, a *many-to-one* referential integrity relationship exists between the **OEID** attribute of the **State_Characteristics** table and the **EID** attribute of the **Events** table insuring that each state characteristic stored in the **State_Characteristics** table is associated with the observation event at which it was recorded.

5.5.7 State Data Tables: State_Numbers, State_Short_Texts, State_Long_Texts

The SPEHR schema has from one to *n* tables to actually store the static attribute values linked to states, one for each native storage type supported by the database management system that is necessary to store static attributes of states. They all take the following fundamental form; examples are displayed in Figure 2 and implemented in the example databases that accompany this work.

The tables named **State_<native storage type name>** contain one row for each specific state-associated static attribute value stored in a SPEHR database. Attributes of the **State_<native storage type name>** tables include:

- the unique identifier of the state characteristic with which each static state attribute value is associated: **SCID**,
- a value for each static state attribute: **State_<native storage type name>**,

One-to-one referential integrity relationships exists between the **SCID** attributes of the **State_<native storage type name>** tables and the **SCID** attribute of the **State_Characteristics** table insuring that each static state attribute value stored in the **State_<native storage type name>** tables is associated with a valid state characteristic stored in the **State_Characteristics** table.

5.6 Memberships

A prominent and often recurring general theme in dynamic models of human populations is *membership*; a relationship that exists between a *collection* and a specific *member* of that collection (See Benzler et al., 1998). Collections (or aggregations) are simply groupings of individual members that persist from the time when there were at least two members of the group until the group dissolves. Examples include marital unions, families (including children), households (including non-family members), homesteads, villages and *places*. A place is a collection of the people who reside there; and the memberships themselves are more specifically *residences* at that location. Because they have a beginning and an end, collections are *states* in the SPEHR sense, as are the members themselves. Furthermore member*ships* also have a beginning and an end making them too *states* in the SPEHR sense. Consequently collections, memberships and members are all stored in the **States** table in a SPEHR-based database. All that is necessary to record the special relationship that these states share with one another is to link them and label their relationship. This is accomplished by adding a table with three attributes that store the identifiers of collection, membership and member states – along with the identifier of the membership type of the membership that links the three state identifiers recorded in each row. It is because the *collection—membership—member* theme is sufficiently common and general that it warrants an addition to the core SPEHR schema.

Two tables are added named **Memberships** and ***Membership_Types***, enclosed in a box labeled “Memberships” in Figure 2.

5.6.1 Table: Membership_Types

The metadata table named ***Membership_Types*** contains one row for each specific type of membership that an individual SPEHR database can understand and record. Attributes of the ***Membership_Types*** table include:

- a unique identifier for each membership type: **MTID**,
- the unique identifier of the member state type associated with each membership type: **MSTID**,
- the unique identifier of the collection state type associated with each membership type: **CSTID**,
- a name for each membership type: **Membership**, and
- a brief description of each membership type: **Description**.

Many-to-one referential integrity relationships exist between the **MSTID** and **CSTID** attributes of the ***Membership_Types*** table and the **STID** attribute of the ***State_Types*** table insuring that each membership type stored in the ***Membership_Types*** table is associated with valid state types stored in the ***State_Types*** table corresponding to the state types of the collection and member states that participate in each membership type stored in the ***Membership_Types*** table.

5.6.2 Table: Memberships

The table named **Memberships** records the links (relationships) between individual collection, membership and member states and contains one row for each membership that is recorded in an individual SPEHR database. Attributes of the **Memberships** table include:

- the unique identifier of the collection state associated with each membership: **CSID**,
- the unique identifier of the membership state associated with each membership: **MPSID**,
- the unique identifier of the member state associated with each membership: **MSID**,
- the unique identifier of the membership type associated with each membership: **MTID**, and
- the unique identifier of the observation event that recorded each membership: **OEID**.

A *many-to-one* referential integrity relationship exists between the **MTID** attribute of the **Memberships** table and the **MTID** attribute of the ***Membership_Types*** table insuring that each membership stored in the **Memberships** table is associated with a valid membership type stored in the ***Membership_Types*** table.

To insure that each membership is associated with one collection, one membership and one member; *many-to-one* referential integrity relationships exist between the **CSID**, **MPSID** and **MSID** attributes of the **Memberships** table and the **SID** attribute of the **States** table. Together these form the primary key of the **Memberships** table.

Last, a *many-to-one* referential integrity relationship exists between the **OEID** attribute in the **Memberships** table and the **EID** attribute of the **Events** table insuring that each membership stored in the **Influences** table is associated with the observation event at which it was recorded.

The **Memberships** table can contain many rows of the same type (for example, type ‘residence’), but each of these must correspond to a unique membership of that type in the real world, and to reflect that each will have a unique combination of IDs in the **CSID**, **MPSID** and **MSID** attributes. A collection and a member can have more than one membership relationship; each with a different membership state with its own unique identifier (and start and stop times), **MPSID**.

5.7 Transitions

Transitions are cousins of memberships. Where a membership explicitly records the state of membership in a collection with a defined beginning and end, a transition explicitly records a change in membership with a defined time when the change occurred – the transition from membership in one collection to membership in another, or from not being a member to being a member or being a member to not being a member. Memberships provide a state-centric way of handling this issue, whereas transitions provide an event-centric alternative. Both options are included in SPEHR because it can be significantly more or less efficient and/or intuitive to use one or the other depending on the reality that must be modeled. With enough messaging they are equivalent, and to keep the conceptual design of SPEHR elegant it would be best to settle on one and leave the other out. But because the author has not found compelling general reasons to favor one or the other they are both presented here.

Within SPEHR *transitions* are conceptualized as special collections of influences. Transitions group together related influences that account for the transition of a member state between two collections. For example imagine a death that results in the breakup or a large household. The numerous members of that household disperse in small groups and either form new households or join other existing households. In this case the households are the collections of interest and the individual people are the members of interest. The various households are linked to the death event with separate influences that indicate that they are either losing or gaining new members as a result of and at the time of the death. Likewise the individual people are linked to the death event with separate influences that indicate that they are leaving and joining households. Both the households and the people are linked to the death through lots of separate influences, but *specific* people are not linked directly to *specific* households, and when more than two households are involved it is not possible to identify from these influences alone to which specific new household each specific person goes. The resolution to this indeterminacy is to group the related influences together so that the household-linked influences that reflect loss and gain of members are linked to the people-linked influences that reflect leaving and joining those households. If we take a single person as an example, there are four related influences: 1) the household of origin-linked influence reflecting the loss of a household member, 2) the destination household-linked influence reflecting the acquisition of a new member, 3) the person-linked influence reflecting the departure from the household of origin, and 4) the person-linked influence reflecting the arrival at the destination household (the latter two can be combined into one influence if desired). Linking these four influences allows a straightforward one-to-one association of the person with the two households involved. Furthermore, this approach is fully general, albeit a little abstract, and can be applied to any transition of this type. What is required from the schema point of view is a table to store the transitions (influence groupings) themselves, and then because there can be a many-to-many relationship between transitions and influences, a separate table to store (potentially many) links between individual transitions and individual influences (one row per link).

Four new tables are added to the core SPEHR schema to handle transitions. The transitions are stored in a table named **Transitions**, and the links between transitions and influences, called transition influences, are stored in a table named **Transition_Influences**. Both the **Transitions** and **Transition_Influences** tables have associated metadata tables that define the domains of transitions and transition influences. All four are enclosed in a box labeled “Transitions” in Figure 2.

5.7.1 Table: Transition_Types

The metadata table named ***Transition_ Types*** contains one row for each specific type of transition that an individual SPEHR database can understand and record. Attributes of the ***Transition_Types*** table include:

- a unique identifier for each transition type: **XTID**,
- a name for each transition type: **Transition**, and
- a brief description of each transition type: **Description**.

5.7.2 Table: Transitions

The table named **Transitions** records the transitions (collections) and contains one row for each transition that is recorded in an individual SPEHR database. Attributes of the **Transitions** table include:

- a unique identifier for each transition: **XID**, and
- the unique identifier of the transition type associated with each transition: **XTID**.

A *many-to-one* referential integrity relationship exists between the **XTID** attribute of the **Transitions** table and the **XTID** attribute of the ***Transition_Types*** table insuring that each transition stored in the **Transitions** table is associated with a valid transition type stored in the ***Transition_Types*** table.

5.7.3 Table: Transition_Influence_Types

The metadata table named ***Transition_Influence_Types*** contains one row for each specific type of transition influence that an individual SPEHR database can understand and record. Attributes of the ***Transition_Influence _Types*** table include:

- a unique identifier for each transition influence type: **TITID**,
- the polarity of the action of each transition influence type (*collect* or *disperse* members): **Collect_Or_Disperse**,
- the role of the state associated with this transition influence type (*member* or *collection*): **Member_Or_Collection**,
- the unique identifier of the influence type associated with this transition influence type: **ITID**, and
- the unique identifier of the state type (type of the state associated with the influence type that is associated with this transition influence type) associated with this transition influence type: **STID**.

A *many-to-one* referential integrity relationship exists between the **ITID** attribute of the ***Transition_Influence_Types*** table and the **ITID** attribute of the ***Influence_Types*** table insuring that each transition influence type stored in the ***Transition_Influence_Types*** table is associated with valid influence type stored in the ***Influence_Types*** table.

Additionally, a *many-to-one* referential integrity relationship exists between the **STID** attribute of the ***Transition_Influence_Types*** table and the **STID** attribute of the ***State_Types*** table insuring that each transition influence type stored in the ***Transition_Influence_Types*** table is associated with valid state type stored in the ***State_Types*** table.

5.7.4 Table: Transition_Influences

The table named **Transition_Influences** records the links (relationships) between individual transitions (collections of influences) and individual influences and contains one row for each transition influence that is recorded in an individual SPEHR database. Attributes of the **Transition_Influences** table include:

- the unique identifier of the transition associated with each transition influence: **XID**,
- the unique identifier of the influence associated with each transition influence: **IID**,
- the unique identifier of the transition influence type associated with each transition influence: **TITID**, and
- the unique identifier of the observation event that recorded each transition influence: **OEID**.

Four referential integrity constrains apply to the **Transition_Influences** table. One, a *many-to-one* referential integrity relationship exists between the **XID** attribute of the **Transition Influences** table and the **XID** attribute of the **Transitions** table insuring that each transition influence stored in the **Transition_Influences** table is associated with a valid transition stored in the **Transitions** table.

Two, a *many-to-one* referential integrity relationship exists between the **IID** attribute of the **Transition Influences** table and the **IID** attribute of the **Influences** table insuring that each transition influence stored in the **Transition_Influences** table is associated with a valid influence stored in the **Influences** table.

Three, a *many-to-one* referential integrity relationship exists between the **TITID** attribute of the **Transition Influences** table and the **TITID** attribute of the ***Transition_Influence_Types*** table insuring that each transition influence stored in the **Transition_Influences** table is associated with a valid transition influence type stored in the ***Transition_Influence_Types*** table.

Last, a *many-to-one* referential integrity relationship exists between the **OEID** attribute of the **Transition Influences** table and the **EID** attribute of the **Events** table insuring that each transition influence stored in the **Transition_Influences** table is associated with the observation event at which it was recorded.

5.8 Processes

A process is a “the series of actions, operations, or motions involved in the accomplishment of an end” (Merriam-Webster, 1996). Human experience is full of such series of causally related events; for example the process of getting married and starting a family may involve finding a spouse, getting engaged, getting married, moving to a new location to settle with the spouse, setting up a new household, and maybe having a baby – with much variation from one couple to the next! The point is that a series of events are linked in a clearly related causal framework, and it would be valuable to be able to record this in a database.

SPEHR accommodates *processes* by stretching the definition of the state to include state types of the general type “process”. After all a process has a beginning (the first event in the series) and an end (the last event in the series) and maintains a state of “in process of” between those two. Additionally appropriate influence types must be defined to link events in a process to the process.

Recording a process in a SPEHR database involves instantiating a state of the relevant process type and then linking events in the process to the instance of the process by instantiating influences of the correct types linked to the relevant events and the process. The “London-New York” example (see section 4.4) has a process that links the death of Richard with the migration of Elizabeth and Beatrice form London to New York, and this process is recorded in the example databases accompanying this work (see section 10).

5.9 Normalization

“Normalization” in database theory describes the formalized process of reducing or eliminating *redundancy* (duplication) of data in a database. A highly normalized database design is one in which very few or no datum are duplicated anywhere in the database. The significant emphasis on normalization in database theory is motivated primarily by the desire to prevent inconsistency among datum that are recorded in multiple locations in a database. Inconsistency or non-correspondence between multiple “copies” of the same data is colloquially termed data “corruption”, leading to a “corrupt” database. Obviously this is a serious problem and it is one of the most important priorities of a database designer to prevent data corruption.

Consequently, a highly normalized design was one of the design criteria for SPEHR, and in particular with relevance to the temporal data stored in SPEHR. To accomplish this, the tables that comprise SPEHR do not duplicate any metadata or primary data among themselves. Critically, *the timestamps that provide the temporal dimension to SPEHR are stored in only one attribute of only one table* – the **Events** table – and it is through unique links to the **Events** table that other tables are given a temporal meaning.

Another advantage of a highly normal form is that edits and corrections to datum are made in one and only one place and automatically propagate through the database via the links to the one place where the edited datum is stored. Important in this regard for SPEHR is the fact that corrections to timestamps (dates) are made in only one place – the timestamp attribute of the **Events** table – thereby preventing the possibility of updating one or two copies of a timestamp and forgetting to update the third and forth (more obscure) copies located somewhere else in the database. It is not possible to create temporal inconsistency of this type in SPEHR.

An aspect of normalization that may have been more important early in the history of relational databases is that reduction or elimination of duplication can reduce the physical storage space necessary to store a database. Given the very large and cheap storage available at this time, this advantage of normalization is no longer as important.

Caution: The SPEHR schema presented in Figure 2 and the example databases based on that schema that accompany this work are not fully normalized. Many table-level unique keys are defined in the example schema and example databases that are not necessary or perhaps desirable in an operational implementation of SPEHR. They are included here to make the examples easier to read and understand, but it is strongly recommended to anyone implementing SPEHR in an operational setting that the operational schema be further normalized to remove unnecessary table-level unique identifiers, such as the **IID** attribute of the **Influences** table.

5.10 Observations

The end product of a longitudinal study always involves relating the occurrence of something to the exposure to that occurrence accumulated by the units of observation. In population and health studies exposure is often measured in “person years” and the object is to quantify the risk of some event controlling for the exposure to the event experienced by people of different types: differing ages, sexes or residence for example. It is also critically necessary to know for all units of analysis the most recent date when they were observed; this being the date after which nothing is known about them and consequently past which their exposure is unknown and therefore past which they cannot be included in the analysis. For these reasons the “period of observation” for each unit of analysis is of critical importance to longitudinal (and/or survival) analysis. Moreover, from an operational point of view it is necessary to monitor and record when (and potentially where) study participants have been contacted, or “observed”.

As a result, observation tracking has been built into the SPEHR schema. This is done through introduction of a special event type: “observation” in the ***Event_Types*** table which is the only piece of metadata that “comes with” SPEHR. To support the linking of various pieces of primary data with the observation event during which they were captured, an **OEID** (observation event ID) attribute appears in many tables. This attribute stores the unique identifiers of the observation events during which the data contained in each row of the table were collected.

The **Events** table itself has an **OEID** attribute to link each event to the observation event during which it was captured. Likewise, the **Influences**, **Transition_Influences**, **Memberships**, **Event_Characteristics** and **State_Characteristics** tables all have an **OEID** attribute to link each row in those tables to the observation event during which the data in the row were captured.

5.11 Event Histories

Another element of longitudinal analysis that featured in the design considerations of SPEHR is the fact that many longitudinal or survival analysis techniques require “event histories” of some sort – essentially chronological lists of events that have occurred to each unit of analysis, including events signifying when the units of analysis come into and go out of observation. Schema designs that distribute events and dates in many separate tables make it difficult (sometimes very difficult) to gather together all the events into a single view that simulates an event history for each unit of analysis. It is in fact when attempting to create event histories that many inconsistencies in duplicated dates often appear (see section 5.9).

The idea of keeping all events in a single table was partly motivated by the necessity to generate event histories, and in practice it is indeed very easy to generate event histories in a SPEHR database. And furthermore with the addition of two tables that define age intervals and calendar intervals (historical periods), it is then a trivial matter to generate age and historical period-specific “person year” files that contain one row for each year that a person lived.

Last, having the ability to easily generate accurate chronological lists of events makes it easy to generate synthetic states that may not be built into a SPEHR database. For example, it may be advantageous to generate interbirth intervals, but states that span the interbirth interval are not defined in the SPEHR database being used. This task is greatly simplified by being able to quickly and easily generate an accurate list of all births for each woman in the database – it is a straightforward piece of SQL then to generate the interbirth intervals, or in SPEHR’s terminology, the interbirth *states*.

5.12 Metadata, Data Dictionaries and *SPEHR Database Sharing*

Together with its reflection of the GTDM, the metadata-driven concept that pervades SPEHR is what sets SPEHR apart from other temporal database designs. The GTDM provides it with a conceptually general and flexible way to represent temporal processes, and the metadata provide it with a general and flexible way to store that representation in a *schema-invariant* relational database. Schemainvariance describes the fact that the logical structure of a SPEHR database does not need to change to expand and accommodate new entities. The two important components of the metadata-driven approach are the metadata themselves and the meaning that they confer to the primary data – the data dictionary function – and the fact that the metadata allow the database schema to remain constant while new entities are added and new meaning is given to the database.

The consequence of the first is that the data dictionary is built into a SPEHR database and the primary data are always accompanied by their definitions; while the consequence of the second is that the logical structure of a SPEHR database does not need to change to add new entities – no new tables or relationships are necessary to expand the reality that a SPEHR database reflects. This allows a SPEHR database to grow gracefully with the project that it supports without developing an impenetrably complicated and unwieldy schema.

Finally and perhaps most importantly, the metadata-driven schema allows different SPEHR database users to effortlessly and transparently share their data with each other. This is possible because the underlying schemas of all SPEHR databases are the same; all that differentiates them are the metadata. Provided two SPEHR databases share some subset of metadata, the primary data that those metadata describe can be pooled into one SPEHR database and managed and/or analyzed as one dataset, and all that is necessary to accomplish this is loading the primary data into a single SPEHR database that already contains their common (shared) subset of metadata.

To make this happen easily only one additional requirement is necessary; namely that someone hold a *master store* of SPEHR metadata that serves as the *standard* metadata for all SPEHR database users who want to be able to easily share data among themselves. The concept of a master metadata store brings another advantage beyond facilitating the sharing and comparing of data between individual SPEHR databases; it also allows SPEHR "modules" developed by individual SPEHR users to be "posted" to the master store from where they can be "downloaded" by other users who wish to quickly and easily add a new module to their own SPEHR database. This facility could greatly expedite the extension and expansion of studies that wish to incorporate a complicated new module but do not have the time or resources to embark on a substantial database upgrade in order to manage the new data.

Together, its generality, its ability to share and compare primary data easily and accurately and its ability to share "modules" makes SPEHR potentially interesting as a standard longitudinal relational database schema. Section 6 below discusses a simple, straightforward addition to the SPEHR schema that enables its ability to manage data from multiple "sites" in a single SPEHR database, and a working example of that schema is discussed in section 6.2 and accompanies this work as a working database.

5.13 The London-New York Example in SPEHR

The example study described in section 4.4 above and in Figure 1 is also implemented as a working example of the SPEHR schema in an accompanying MS Access 2000 database named "SPEHR-London-NewYork.mdb". The SPEHR schema presented in Figure 2 is implemented exactly and the metadata necessary to reflect the London-New York example study are inserted, including those presented in Table 1 (in the ***Influence_Types*** metadata table). To open the example database, you must have a version of MS Access able to open "Access 2000" version Access databases running on your computer. When you have that, simply double-click on the database file to open and begin working with it.

In addition to the tables and relationships necessary to implement the SPEHR schema, there are a number of example queries (views written in SQL) defined in the example database to demonstrate that it is easily feasible to view data in a SPEHR database in ways that are more familiar. The queries have self-explanatory names like: "Select_Life_History_Person", "Select_Current_Population" and "Select_Person_Names_Sexes_Vital_Dates".

6 Multi-Site Structured Population Event History Register

As research questions become more complex and seek measurements on less common events, it is natural to consider expanding single site studies to encompass multiple sites and hence multiple populations (with larger numbers). This leads directly to the need to combine data from several existing sites, or to initiate new research at a number of different sites; both requiring the interdigitation of potentially different data management systems and/or the creation of a new data management system that can handle data from all the sites simultaneously. Study designs of this type are increasingly common in vaccine behavioral disease prevention trials being conducted in the developing world (See for example: HVTN, 2004; IAVI, 2004; SAAVI, 2004).

Another methodology currently being developed is so-called *sample vital registration*, conceived to substitute for full vital registration in parts of the world where full vital registration is lacking (MEASURE, 2004a, b). Sample vital registration is basically DSS-lite covering much larger geographic areas of a country and designed to cover enough of all parts of a country to produce representative data at a national level, but *without much of the detail* typically associated with a full DSS. Obviously this creates a requirement that data from several different regional populations be managed together, and it is highly conceivable that as time progresses each region will want to add to and specialize the data that it collects in order to more finely focus on the issues that pertain in that region and not in others. Furthermore, a suggested enhancement to this methodology is to place one or more full DSSes in each region in order to provide the region with much deeper and fuller longitudinal information on a smaller population that may be representative of the region. The result of a combined DSS-sample vital registration system would be a comparatively inexpensive basic data collection platform that is both nationally representative and detailed enough at the regional level to provide a means of conducting in depth studies and monitoring at a penetrating level of detail. Such a system would require a highly flexible data management facility that is able to incorporate heterogeneous data from different regions, different data collection systems and different historical periods and grow gracefully with the substantive requirements of the system for a long period of time.

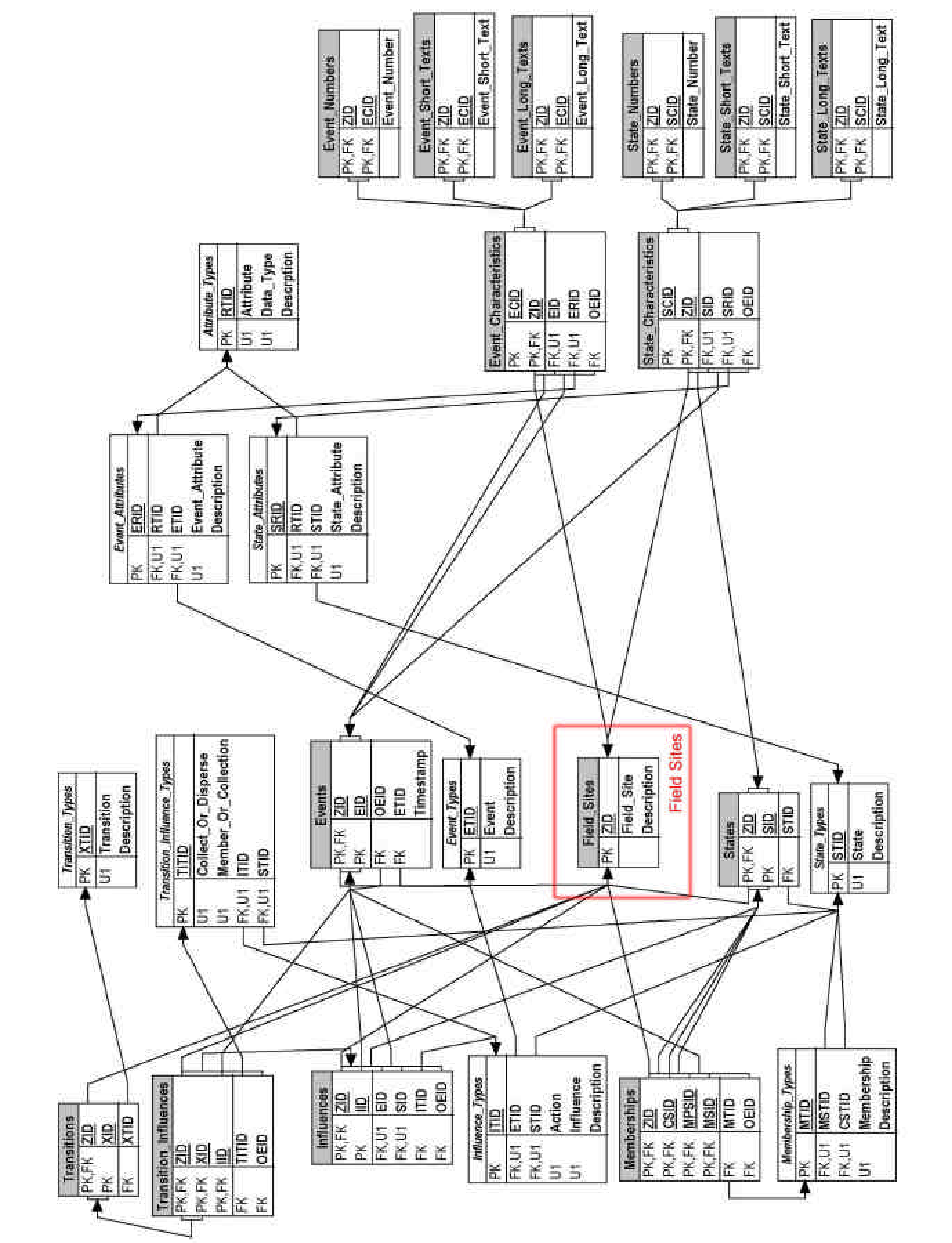
As alluded to in section 5.12, the potential for a SPEHR-based database to easily and efficiently manage data from many sites in one schema is one of SPEHR's most significant attractions. This section briefly introduces the *multi-site* version of SPEHR that is able to do that.

6.1 Multi-Site SPEHR Schema

In order to easily manage data from multiple sites or projects, a straightforward addition is necessary to the existing single-site SPEHR schema. One table named **Field_Sites** is added to contain a list of all the field sites contributing data to the multi-site SPEHR database. Each site is assigned a unique identifier, and that identifier is used to differentiate the primary data from each site from the primary data from other sites in all the primary data tables in SPEHR.

The metadata tables remain unchanged and contain exactly the same type of metadata as they do in the single-site version of SPEHR. However, it is when combining data from many sites that the metadata become critical, and the central metadata store (see section 5.12) plays a key role. Within each metadata table the unique identifiers associated with each row must be consistent across all SPEHR databases contributing data to the central multi-site SPEHR database. That is, the metadata that describe a specified type of a general object must have the same unique identifier in all of the SPEHR databases contributing data. That way when the data are all merged the metadata from each database will mean the same thing and provide the same meaning to the primary data coming from each database – obviously each metadata table row will occur only once in the combined database. For these reasons it is critical to manage the uniqueness and consistency of the metadata and that is why the central metadata store is so important. Moreover, it is unlikely that all the contributing databases will have the same metadata specification because they will have been individualized to some extent; the standardized metadata specification will allow the importation of only those primary

Figure 3: Entity Relationship Diagram of Multi-Site SPEHR



data that are compatible and desired from the contributing databases.

Figure 3 contains an entity-relationship diagram of the multi-site SPEHR schema. The **Field_Sites** table is enclosed in a box labeled “Field Sites”.

6.1.1 Table: Field_Sites

The table named **Field_Sites** contains one row for each field site that an individual multi-site SPEHR database records. Attributes of the **Field_Sites** table include:

- a unique identifier for each field site: **ZID**,
- a name for each field site: **Field_Site**, and
- a brief description of each field site: **Description**.

6.1.2 Primary Data Tables: Adding the **ZID** Attribute

In order to differentiate data from each site in each primary (not metadata) table, an additional attribute is added to each primary data table to store the unique identifier of the field site from which the data in each row comes, the value in the **ZID** attribute of the **Field_Sites** table that corresponds to the field site from which the data come. These new attributes can be seen in Figure 3.

Additionally, existing referential integrity relationships between primary data tables that contain the **ZID** attribute are joined by additional referential integrity relationships that require matching on the **ZID** attributes as well.

6.2 The London-New York and Johannesburg-Durban Examples in Multi-site SPEHR

As with the single-site version of SPEHR, a working example of the multi-site version has been created and accompanies this work as an MS Access 2000 database named “SPEHR-FINAL-Merged.mdb”. This database is an implementation of the schema displayed in Figure 3. The sites that contribute to this example database are the “London-New York” example site discussed in sections 4.4 and 5.13 and displayed in Figure 1 and another example site called “Joburg-Durban”. The Joburg-Durban example is very similar to the London-New York example, differing only in place names and times. An example single-site SPEHR database of the Joburg-Durban example also accompanies this work named “SPEHR-Joburg-Durban.mdb”. The contents of the London-New York and Joburg-Durban single-site SPEHR databases are combined according to the procedure described above in 6.1 to yield the multi-site example database.

7 Further Components of SPEHR

Three major additional components would greatly enhance SPEHR’s ability to support an efficient, reliable, useful production data management system for a longitudinal study of human populations, and ongoing work by the author is addressing these. They are:

1. a generalized metadata-driven facility to model and store both temporal and non-temporal integrity constraints on primary data,
2. a generalized metadata-drive facility to model and store questionnaire or other data capture instruments, and
3. a generalized suite of commonly required views and extraction tools that would be useful with reference to many or all different types of states stored in a SPEHR database.

The first is critical to maintaining the consistency and integrity of the data. Ordinary referential integrity constraints insure that relationships between individual tables are maintained through maintenance of primary and foreign key relationships, but they do little or nothing to provide a standardized and easy to use way of insuring temporal integrity. Temporal integrity refers to a situation in which all events are correctly sequenced, states that should not overlap in time do not, states that should overlap in time do, states that should abut or meet each other in time do, and states that should not "meet" do not, etc. Essentially temporal integrity insures that facts stored in a temporal database are associated with valid timestamps, correctly sequenced chronologically and potentially associated with well formed states that are in the correct sequence and relationship to other states in the database, see Date, Darwen and Lorentzos (2002a; 2002b). Because SPEHR is built around a clear conceptual standard and implemented in a static schema, it will be possible to develop conceptually general standard methods for assessing and enforcing temporal integrity.

The second extends the usefulness of SPEHR in an important way and allows the joint, concurrent management of the data and the instruments that capture them together in one database. This allows the primary data to be permanently linked to the instrument and specific question that captured them, thus significantly extending the data dictionary to include the full context in which each datum was collected. Managing the instruments and data together also allows the instruments to make use of the stored data to perform real-time validity checks on data that are being captured and to flag or reject potentially false or inconsistent data while it is being captured. Finally, such a system can be adapted to work on hand held computers that allow field workers to go into the field with true *smart* questionnaires that make full use of existing data to improve the quality of newly captured data and potentially to save a lot of time in the capture process as well.

The third is not really a component on its own but rather an important sub-component of all components of SPEHR. SPEHR's generalized, static, metadata-driven schema makes it possible to define general, metadata-driven routines that actively interact with the database in many different ways. For example, integrity checks of various types may make use of generalized routines that identify overlapping states of whatever type necessary or miss-sequenced events whether they belong to people, households or whatever. For analysis, general metadata-driven routines could be developed to calculate various measures of exposure. So instead of having to write separate routines to do similar things, it will be possible to write general routines that make use of SPEHR's metadata and non-changing schema to accomplish the same thing with reference to many different types of the underlying objects.

These three additions are conceptualized within the same generalized metadata-driven framework to function in ways very similar to the existing SPEHR schema and to interact with it at a deep level, sharing metadata and providing new metadata to enhance the integrity and meaning of data stored in the existing SPEHR schema.

8 Discussion

The work presented here is built around the philosophy that generalization and standardization are greatly worth attaining, in the service of improving efficiency, accuracy, reliability and comparability. To realize a high degree of generalization and standardization it is necessary to develop basic, abstract representations of the real world and to identify the underlying similarities and congruencies among the entities we wish to manipulate – model, capture, store and retrieve. The General Temporal Data Model provides the general abstract representation of temporal reality that we need to represent the inter-related histories of various entities as time progresses. The Structured Population Event History Register provides a relational schema for implementing a working version of the GTDM in a relational database management system; one that is designed specifically to capture the inter-related histories of human beings. Building on that the multi-site version of SPEHR allows data describing the inter-related histories of people living in different populations and captured by different projects to be managed and manipulated together in one SPEHR-based database.

SPEHR is a flexible tool that allows a user to easily define the structured temporal data that they want to store and manipulate, to actually store and manipulate that data, and to refine and re-define the definitions of the data as time goes on – all without making changes to the schema of the relational database that implements SPEHR. As such SPEHR is not a "database for DSS" or a database for anything else in particular; to become a DSS database or a database for project X, a specific set of metadata must be defined and stored within SPEHR to allow it to store and manipulate the data collected by a DSS, or by project X. It is worth stressing that the GTDM is sufficiently general to model the histories, inter-related or not, of a great many different kinds of “things”. Populations of animals (ecology studies) come to mind immediately as do populations of hospital patients, populations of registered automobiles, inventories, populations of on-going jobs in a large organization and populations of calendar appointments, to name a few. There may be further application of the GTDM and SPEHR beyond population and health studies.

A common comment received from colleagues who have examined SPEHR is that the GTDM objects are so general that it is very difficult to conceptualize how to reorganize the data into more familiar forms. Although this observation is cogent, it cuts two ways. Relational databases that store any meaningful level of temporal information describing human populations are all very complex and take a lot of effort to understand and manipulate. Although the SPEHR schema is more abstract and perhaps more “tricky” than most, it has as its foundation a *few* simply consistent concepts, and once those are mastered, there is nothing else to learn about a how to interact with a SPEHR database. Once you “get” the basic idea you are set; you will not have to learn the concepts behind the next big revision of your database because the basic concepts behind a SPEHR database will not change. The author has constructed detailed genealogies and other substantially derivative data extracts from data describing human populations stored in a SPEHR database without undue difficulty.

Ongoing work addresses 1) the need to formalize integrity constraints in a general, metadata-drive way within SPEHR, 2) the need to develop and implement a general metadata-drive model of data capture instruments within the SPEHR schema, and 3) to develop general routines to support integrity checking, data manipulation, and extraction of highly manipulated data from SPEHR databases.

Building on the work presented here, two related pieces of work are envisioned for the near future: 1) a detailed SPEHR metadata module (values for all the metadata tables in a SPEHR schema) that is sufficient to describe a “core” DSS SPEHR database, and 2) a functioning implementation of the core SPEHR schema in a production relational database management system such as MS SQL Server or IBM DB2 that incorporates the core DSS metadata and provides an SQL DDL script to create the database.

9 Acknowledgements

The work presented here from experience in data collection and data management in the context of long term field sites studying population and health issues. The author wishes to acknowledge and thank strong long term supporters of this work including Jane Menken, Samuel Preston, Thayer Scudder and James Lee. Many useful discussions and helpful comments have come from Justus Benzler, Kobus Herbst, Bruce MacLeod, Mark Collinson and many others. This work has been supported, in part, by NIA (of the NIH) grants R37 AG10168, 3 R37 AG10168-09S2, 30 AG17248 and 2 P30 AG17248-03. The work presented here greatly expands on Part 4 of the author’s Ph.D. dissertation (Clark, 2001b).

10 Appendix

The appendix contains three example databases discussed in sections 5.13 and 6.2. As discussed in those sections these database are Access 2000 version MS Access databases, each in its own “.mdb” file. To access the databases make sure you have a version of MS Access that can open an Access 2000 version Access database and then simple double-click on the database file or open it from within MS Access. Each example database contains a number of tables and queries that can be opened and manipulated. Referential integrity relationships are also defined and can be viewed using the “relationship view” that is accessed through clicking on the relationship window icon (three connected boxes) in the toolbar at the top of the Access main window.

11 References

Allen, J. F. 1983. "Maintaining Knowledge about Temporal Intervals." *Communications of the ACM*, 26(11 November 1983):832-843.

Allen, J. F. and G. Ferguson. 1994. "Actions and Events in Interval Temporal Logic." *Journal of Logic and Computation*, 4(5):531-579.

Alphora. 2004. "Dataphor: Application Development Toolset". http://www.alphora.com. Accessed: 2004-05-20.

Axinn, W. G., J. S. Barber, and D. J. Ghimile. 1997. "The Neighbourhood History Calender: A Data Collection Method Designed for Dynamical Multilevel Modeling." Pp. 355-392 in Sociological Methodology, edited by A. E. Raftery. Cambridge Massachusetts: Blackwell Publishers.

Benzler, J. and S. J. Clark. Under Review 2004. "Towards a Unified Timestamp with Explicit Precision." *Demographic Research*.

Benzler, J., K. Herbst, and B. MacLeod. 1998. "A Data Model for Demographic Surveillance Systems". www.indepth-network.org/publications/DM_for_Demographic.htm. Accessed: June 2003.

Binka, F., P. Ngom, J. Phillips, K. Adazu, and B. MacLeod. 1999. "Assessing Population Dynamics in a Rural African Society: the Navrongo Demographic Surveillance System." *Journal of Biosocial Science*, 31(3):375-391.

Binka, F. N., K. Adazu, M. Adjuik, L. A. Williams, C. Lengeler, G. H. Maude, G. E. Armah, B. Kajihara, J. H. Adiamah, and P. G. Smith. 1996. "Impact of Impregnated Bednets on Child Mortality in Kassena-Nankana District, Ghana: A Randomised Controlled Trial." *Tropical Medicine and International Health*, 1:147-154.

Clark, S., E. Colson, J. Lee, and T. Scudder. 1995. "Ten Thousand Tonga: A Longitudinal Anthropological Study from Southern Zambia: 1956-1991." *Population Studies*, 49:91-109.

Clark, S. J. 2001a. An Investigation into the Impact of HIV on Population Dynamics in Africa. Ph.D. dissertation in Demography. Philadelphia, Pennsylvania: University of Pennsylvania.

—. 2001b. "Part 4: The Structured Population Event History Register - SPEHR." Pp. 356-378 in An Investigation into the Impact of HIV on Population Dynamics in Africa, Ph.D. dissertation in Demography. Philadelphia, Pennsylvania: University of Pennsylvania.

Cliggett, L. 1997. My Mother's Keeper: Changing Family Support Systems for the Elderly in the Gwembe Valley, Zambia. Ph.D. dissertation in Anthropology. Bloomington, Indiana: Indiana University.

Colson, E. 1960. Social Organization of the Gwembe Tonga. Manchester: Manchester University Press.

—. 1964. "Social Change and the Gwembe Tonga." *Human Problems in British Central Africa*, 35:1-13.

—. 1971. The Social Consequences of Resettlement: The Impact of the Kariba Resettlement upon the Gwembe Tonga. Manchester: University of Manchester Press.

Colson, E. and T. Scudder. 1987. For Prayer and Profit: The Ritual, Economic and Social Importance of Beer in Gwembe District, Zambia, 1950-1982. Stanford: Stanford University Press.

Date, C. J. 2000. "Chapter 1: An Overview of Database Management." Pp. 2-32 in An Introduction to Database Systems. Reading Massachusetts: Addison-Wesley.

Date, C. J., H. Darwen, and N. A. Lorentzos. 2002a. "Chapter 11: Integrity Constraints I: Candidate Keys and Related Constraints." Pp. 187-212 in Temporal Data and the Relational Model. San Francisco: Morgan Kaufmann.

—. 2002b. "Chapter 12: Integrity Constrains II: General Constraints." Pp. 213-244 in Temporal Data and the Relational Model. San Francisco: Morgan Kaufmann.

—. 2002c. Temporal Data and the Relational Model. San Francisco: Morgan Kaufmann.

Desgrees du Lou, A., G. Pison, and P. Aaby. 1995. "Role of Immunisations in the Recent Decline in Childhood Mortality and the Changes in the Female/Male Mortality Ratio in Rural Senegal." *American Journal of Epidemiology*, 142:643-652.

Etzion, O., S. Jajodia, and S. Sripada, Editors. 1998. Temporal Databases: Research and Practice, vol. 1399, Lecture Notes in Computer Science, Edited by G. Goos, J. Hartmanis, and J. van Leeuwen. Berlin: Springer.

Forster, D. and R. W. Snow. 1995. "An Assessment of the Use of Hand-Held Computers During Demographic Surveys in Developing Countries." *Survey Methodology*, 21:193-199.

Forster, P. G. 1995. "Anthropological Studies of Kinship in Tanzania." Pp. 70-117 in Gender, Family and Household in Tanzania, edited by C. Creighton and C. K. Omari. Brookfield: Avebury.

Garenne, M. 1995. "Do Women Forget Their Births? A Study of Maternity Histories in a Rural Area of Senegal (Niakhar)." *Population Bulletin of the United Nations*, 1994(37/38):43–55.

Garenne, M., C. Becker, and R. Cardenas. 1992. "Heterogeneity, Life Cycle, and the Potential Demographic Impact of AIDS in a Rural Area of Africa." Pp. 269-282 in Sexual behaviour and networking: anthropological and socio cultural studies on the transmission of HIV, edited by T. Dyson. Liege: Editions Derouaux Ordina.

Garenne, M. and P. Cantrelle. 1998. "Three Decades of Reseach on Population and Health: The ORSTROM Experience in Rural Senegal: 1962-1991." in Prospective Community Studies in Developing Countries, edited by D. Gupta, Aaby, Garenne, and Pison. Oxford: Oxford University Press.

Garenne, M., R. Sauerborn, A. Nougtara, M. Borchet, J. Benzler, and J. J. Diesfield. 1997. "Direct and Indirect Estimates of Maternal Mortality in Rural Burkina Faso." *Studies in Family Planning*, 28:54-61.

Gray, R. H., D. Serwadda, M. Med, M. J. Wawer, N. Sewankambo, L. Paxton, F. W. Mangen, C. L. Noah Kiwanuka, D. McNaim, G. Kigozi, and J. Konde-Lule. 1997. "Reduced Fertility in Women with HIV Infection: A Population-Based Study in Uganda." Proceedings of *The Socio-Demographic Impact of AIDS in Africa*. International Union for the Scientific Study of Population. 3-6 February 1997.

Gray, R. H., M. J. Wawer, D. Serwadda, and others. 1998. "Population-Based Study of Fertility in Women with HIV-1 Infection in Uganda." *Lancet*, 351:98-103.

Gulutzan, P. and T. Pelzer. 1999. SQL-99 Complete, Really. Lawrence, Kansas: R&D Books.

HVTN. 2004. "HIV Vaccines Trial Network - HVTN". http://www.hvtn.org/. Accessed: 2004-05-20.

IAVI. 2004. "International AIDS Vaccine Initiative - IAVI". http://www.iavi.org/. Accessed: 2004-05-20.

ICDDR-B. 2004. "ICDDR,B: Centre for Health and Population Research". www.icddrb.org. Accessed: 2004-05-20.

INDEPTH Network. 1998. "INDEPTH Announcement". http://www.indepth-network.org/core_documents/announcement.htm. Accessed: 2004-05-20.

—. 2004a. "INDEPTH Demographic Surveillance Sites". www.indepth-network.org/dss_site_profiles/dss_sites.htm. Accessed: 2004-05-20.

—. 2004b. "An International Network of Field Sites with Continuous Demographic Evaluation of Populations and Their Health in Developing Countries - INDEPTH". www.indepth-network.net; www.indepth-network.org. Accessed: 2004-05-20.

International Organization for Standardization. 2000. "ISO 8601:2000 Representation of Dates and Times." Geneva, Switzerland: International Organization for Standardization.

Jensen, C. S. 2000. Temporal Database Management. Ph.D. dissertation in Department of Computer Science,. Aalborg, Denmark: Aalborg University.

Jensen, C. S., C. E. Dyreson, M. Bohlen, J. Clifford, R. Elmasri, S. K. Gadia, F. Grandi, P. Hayes, S. Jajodia, W. Kafer, N. Kline, N. Lorentzos, Y. Mitsopoulos, A. Montanara, D. Nonen, E. Peressi, B. Pernici, J. F. Roddick, N. L. Sarda, M. R. Scalas, A. Segev, R. T. Snodgrass, M. D. Soo, A. Tansel, P. Tiberio, and G. Wiederhold. 1998. "The Consensus Glossary of Temporal Database Concepts - February 1998 Version." in Temporal Databases: Research and Practice, edited by O. Etzion, S. Jajodia, and S. Sripada. Berlin: Springer.

Lamb, W. H., F. A. Foord, C. M. Lamb, and R. G. Whitehead. 1984. "Changes in Maternal and Child Mortality Rates in Three Isolated Gambian Villages over Ten Years." *Lancet*, 2:912-914.

Linder, F. E. 1971. "The Concept and the Program of the Laboratories for Population Statistics." *Laboratories for Population Statistics Scientific Series*, 1.

MacLeod, B. B., J. F. Phillips, and F. N. Binka. 1996. "Sustainable Software Technology Transfer: The Household Registration System." Pp. 302-310 in Encyclopedia of Library and Information Science, vol. 58, edited by A. Kent. New York: Marcel Dekker.

MEASURE. 2004a. "MEASURE Evaluation: Monitoring and Evaluation to Assess and Use Results". http://www.cpc.unc.edu/measure/. Accessed: 2004-05-20.

—. 2004b. "MEASURE Sample Vital Registration with Verbal Autopsy (SAVVY)". http://www.cpc.unc.edu/news?wid=455&func=viewSubmission&sid=308. Accessed: 2004-05-20.

Merriam-Webster. 1996. "Merriam-Webster's Collegiate Dictionary, Tenth Edition with Merriam-Webster's Collegiate Thesaurus for Windows 95". Accessed: 2004-05-20.

Phillips, J. F., B. MacLeod, and B. Pence. 2000. "The Household Registration System: Computer Software for Rapid Dissemination of Demographic Surveillance Systems." *Demographic Research*, 2. www.demographic-research.org/Volumes/Vol2/6.

Pison, G., A. D. d. Lou, and A. Langaney. 1997. "Bandafassi: A 25-Year Prospective Community Study in Rural Senegal (1970-1995)." Pp. 235-275 in Prospective Community Studies in Developing Countries, edited by D. Gupta, Aaby, Garenne, and Pison. Oxford: Claredon Press, Oxford University Press.

Poulsen, A. G., P. Aaby, H. Jensen, N. Naucler, I. M. Lisse, F. Dias, and M. Melbye. 1997. "Nine Years HIV-2 Associated Mortality in a Community Study from the Guinea-Bissau." *Lancet*, 349:911-914.

SAAVI. 2004. "South African AIDS Vaccine Initiative - SAAVI". http://www.saavi.org.za/. Accessed: 2004-05-20.

Scudder, T. 1962. Ecology of the Gwembe Tonga. Manchester: Manchester University Press.

—. 1985. A History of Development of the Zambian Portion of the Zambezi Valley and the Kariba Lake Basin, Institute for Development Anthropology Working Paper 22. Binghamton: Institute for Development Anthropology.

Scudder, T. and E. Colson. 1977. "Long-Term Field Research in Gwembe Valley, Zambia." Pp. 227-254 in Long-Term Field Research in Social Anthropology, edited by F. G. New York: Academic Press.

—. 1980. Secondary Education and the Formation of an Elite: The Impact of Education on Gwembe District, Zambia. New York: Academic Press.

Shamebo, D., L. Muhe, A. Sandstrom, and S. Wall. 1991. "The Butajira Rural Health Project in Ethiopia: Mortality Pattern of Under Fives." *Journal of Tropical Pediatrics*, 37:254-261.

Shamebo, D., A. Sandstrom, L. Muhe, L. Frij, I. Krantz, G. Lonnberg, and S. Wall. 1993. "The Butajira Rural Health Project in Ethiopia: A Nested Case-Referent Study of Under-five Mortality and its Public Health Determinants." *Bulletin of the World Health Organization*, 71:389-396.

Snodgrass, R. T. 2000. Developing Time-Oriented Database Applications in SQL. San Francisco: Morgan Kaufmann Publishers.

Snodgrass, R. T., M. H. Bohlen, C. S. Jensen, and A. Steiner. 1998. "Transitioning Temporal Support in TSQL2 to SQL3." in Temporal Databases: Research and Practice, edited by O. Etzion, S. Jajodia, and S. Sripada. Berlin: Springer.

Spaccapietra, S., C. Parent, and E. Zimanyi. 1998. "Modeling Time from a Conceptual Perspective." Proceedings of *7th International Conference on Information and Knowledge Management*. Bethesda, Maryland. 2-7, November.

Tollman, S., K. Herbst, M. Garenne, J. S. S. Gear, and K. Kahn. 1999. "The Agincourt Demographic and Health Study: Site Descriptions, Baseline Findings and Implications." *South African Medical Journal*, 89:858-64.

Tollman, S. M. and A. B. Zwi. 2000. "Health system reform and the role of field sites based upon demographic and health surveillance." *Bulletin of the World Health Organization*, 78(1):125-134.

Wyon, J. B. and J. E. Gordon. 1971. The Khana Study: Population Problems in the Rural Punjab. Cambridge, Massachusetts: Harvard University Press.