An Information Architecture to Support the Visualization of Personal Histories

*Catherine Plaisant and Ben Shneiderman**

Human Computer Interaction Laboratory
University of Maryland Institute of Advanced Computer Studies
*Department of Computer Science & Institute for Systems Research
College Park, MD 20742
tel: +1 (301) 405-2768, fax: +1 (301) 405-6707
{plaisant, ben} @cs.umd.edu, http://www.cs.umd.edu/projects/hcil

Abstract: This paper proposes an information architecture for personal history data and describes how the data model can be extended to a runtime model for an intuitive visualization using graphical timelines. Our information architecture, is developed for medical patient records, but is usable in other application domains such as juvenile justice or personal resumes. Our model of personal history data groups events into aggregates which are contained in facets (e.g. doctor visits, hospitalizations, or lab tests). Links enable representation of arbitrary relationships across events and aggregates. Data attributes such as severity can be mapped by data administrators to visual attributes such as color and line thickness. End-users have powerful controls over the display contents and they can modify the mapping to fit their tasks.

Keywords: LifeLines, personal histories, medical patient record, information visualization, graphical user interfaces

Introduction

We all have personal stories to tell. Some people keep diaries, assemble scrapbooks and photo albums, or write autobiographies. Some personal record keeping is more business-like, such as financial transactions (checkbook registers or stock trades), educational histories (transcript of courses), or medical records (blood pressure readings or diet notes). Each person keeps these precious personal records in idiosyncratic formats and annotates them in distinctive and diverse ways. These records could be kept on a computer in generic word processor, spreadsheet, or web page forms to support casual browsing or more structured searching such as by keyword.

When personal history records grow large or become institutionalized there is a need for more orderly record keeping with standard terminology, notations, units, and display formats. While many organizations keep personal data, such as medical, banking, life insurance, and educational records, there is little discussion of how to organize information on personal histories, or how to display this information for rapid and accurate comprehension.

The predominant data modeling schemes are based on the Entity-Relationship Model which provides a barely adequate basis for representation of personal histories. It needs to be extended to describe the semantics of personal histories and to support temporal queries, information presentation, and visualization. An Entity-Relationship Diagram and a

corresponding relational schema is a reasonable environment for making some queries, but it needs additional facilities to specify and extract appropriate overviews of a patient's history, a loan applicant's financial status, or a potential employee's resume.

We employ a series of progressively more complex Entity-Relationship Diagrams and relational schemata to represent personal history summaries. We use these two techniques in an informal manner as they are familiar representations. Our goal is to support the personal history visualization technique called LifeLines [Plaisant et al., 1996] in which an overview of a personal history is seen in a single screen as a series of timelines. Users can filter out unwanted information, zoom in on areas of interest, highlight related information, and get details-on-demand.

Even though this paper does not deal with privacy protection, our information architecture accommodates this crucial aspect of personal history systems. Access restrictions, audit trails and encryption methods have to be considered, and querying has to be possible without divulging the personal identification data.

Previous work on visualizing personal histories has occasionally focused on the medical domain [Kilman and Forslund, 1997]. Cousins and Kahn [1991] present a formal system for representing medical temporal data as events on timelines that can be summarized ("abstracted"). They also describe an interactive environment for visualizing patient data, but a limited model of how to characterize an entire patient history. Other temporal data presentations [Allen, 1983; Sanderson, 1992] have dealt with project management (e.g. Microsoft Project) and monitoring electronic equipment, for example, Karam [1994] proposes a method to describe any timeline generators through the definition of a set of mathematical functions. Hibino and Rundensteiner [1997] describe a visual query language for identifying temporal trends in video data. Allen [1995] describes the use of interactive timelines as an interface to information systems of historical events. He introduces implicit and explicit links between events. Kumar and Furuta [1997] propose a two layer object-oriented model of interactive timelines, namely a content layer and a display layer, based on a structured document model, but their solutions are not specifically tied to personal histories.

We begin with a brief example using Leonardo da Vinci, then focus on medical personal history and the corresponding data model. Finally we propose a compact and convenient data file format for transferring summary data such as LifeLines display data, and discuss the use of preference control panels for the visual presentation.

Sources of personal history data

Personal history data will, in most, cases be exported from a database, but it can also be generated by users with a data entry form or authoring tool (e.g. when entering medical history in a doctor's office or insurance history when applying for a new car insurance policy). Finally, the personal history data could be stored in a credit card-size device for portability and emergency use.

The personal history data can then be displayed in variety of ways: a simple textual chronology, multiple tables grouping events by types, or via a graphical display method such as a Perspective Wall [Mackinlay et al. 1991], or LifeLines.

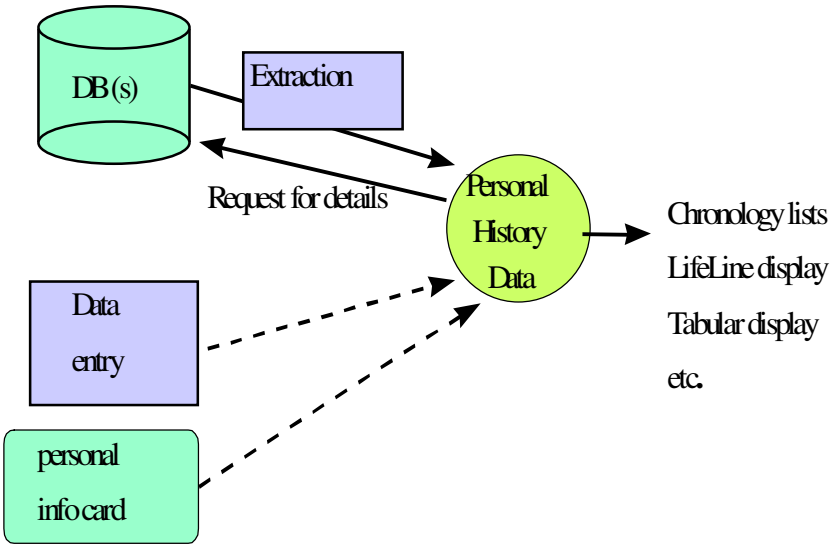


Figure 1: Possible sources of personal history data

Schema 1: Events only (points and intervals)
A simple representation of a personal history would be as a sequence of point events that have a date or interval events that have a range of dates. Point events include birth, death, marriage, and graduation, while interval events include the time they lived in a certain city, held a job, attended schools, or dealt with medical problems. Information presentations are usually organized by sequences of intervals: educational records show semesters; resumes show spans of time; and stock portfolios indicate ranges of dates. As an example we look at a simplified biography of Leonardo da Vinci:

Biography: Leonardo da Vinci, Male, painter
1452 Birth in Vinci, Italy, Verrochio
1473-1474 Paints Ginevra de Benci, oil
1482 Travels to Milan, Italy
1495-1478 Paints Last Supper, fresco
1500 Travels to Florence, Italy
1504-1504 Paints Mona Lisa, oil
1509-1510 Paints St. Anne, oil
1517 Travels to France
1519 Death in Amboise, France, Frere Sebastien

Similar lists are found in the *chronology* sections of most biographical books.

This simple representation is a good starting point, and with tens of thousands of events in a life, it is useful to have multiple event types. For example, in Leonardo's life events might be categorized into Life-Cycle, Paints, and Travels. The database administrator would prepare a set of event types and specify their attributes and permissible attribute

values. This could be shown as an Entity-Relationship Diagram (Figure 2.) When it comes to specifying the visual display properties, the database administrator might select default shape, size, and color attributes for the event markers.

The personal history users would be able to selectively display event types and have the instances appear on the display. The user would be able to look at the early or late stages of a person's life, or focus on events or intervals that occurred on a specific date. Users would be able to reset the control panel to alter the shape, size, and color coding to suit their needs.

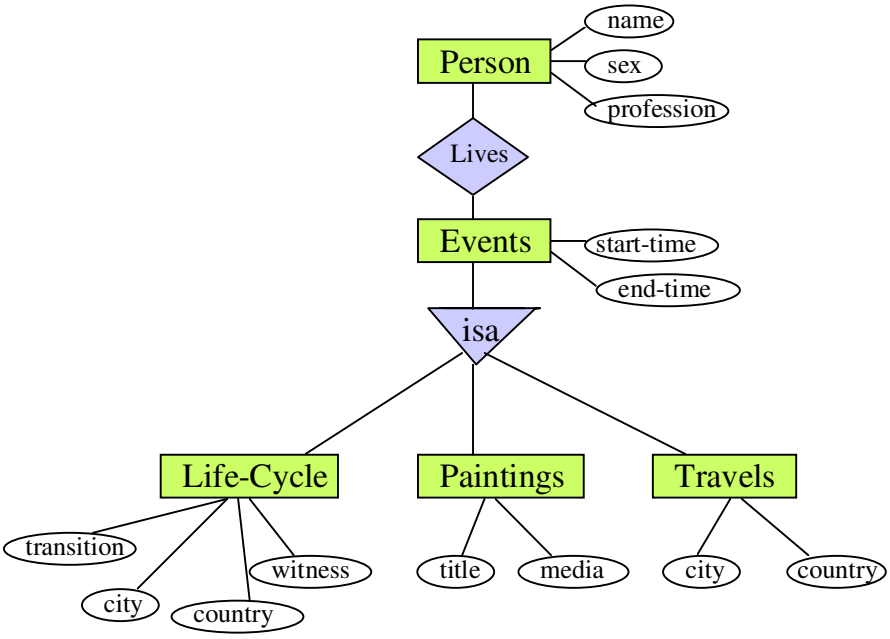


Figure 2: ERD for Schema 1: events only (points and intervals) with Leonardo example

Writing a query to produce the above biography would be relatively simple, but would take days of training to learn the query language and then minutes to compose the query. Form fill-in or other interface approaches might be applied constructively to simplify the learning and query process.

The model can be converted to a relational schema for storage and querying on a database of millions of people. We assume that point events only have a single time (i.e. a START-TIME but no STOP-TIME) and interval events have a START-TIME and END-TIME. We use all capital names to indicate specified terms that will be used in a data definition:

Person (NAME, sex, profession)
Life-Cycle-events (START TIME, END-TIME, transition, city, country, witness)
Paints-events (START-TIME, END-TIME, title, media)
Travels-events (DATE, city, country)

This is a good start, but the richness of human life and the complexity of the questions that users may have means that a more elaborate model is necessary.

Schema 2: Events and facets

As the number of events and intervals grows large it becomes increasingly difficult to browse or review a personal life history. Individuals or decision-makers often segment personal data into facets such as educational, financial, and medical, or into further specifics such as physician visits, hospitalizations, medications, and lab tests. We call these *facets.* Small records may have a simple non-hierarchical list of facets while large and complex records will require a hierarchy of facets. For an academic resume, facets might be the research topics, journal articles, conference papers, grants, courses taught, positions held, and lectures given.

Facets are permanent and do not change over a person's life. For end-users, facets are a natural level to assign access privileges. For example end-users might decide to make their educational facet readable by all, but restrict the medical facet for their physicians. They might elect to protect the financial facet with an additional password. Of course access management is a much larger problem; it will require restrictions to be set at any level of the data model down to a single event to allow complex context sensitive rules to be applied.

Our ERD (Figure 3) shows that a person has multiple facets and that each facet has multiple event types. However, an event can occur in only one facet, so we assign an attribute with the value of its FACET:

Person (NAME, attrP1, attrP2,...)
A-events (ID, START-TIME, STOP-TIME, FACET, attrA1, attrA2,...)
B-events (ID, START-TIME, STOP-TIME, FACET, attrB1, attrB2,...)
C-events (ID, START-TIME, STOP-TIME, FACET, attrC1, attrC2,...)

Users can then view each facet separately, as well as limit data to a specific date or range of dates.

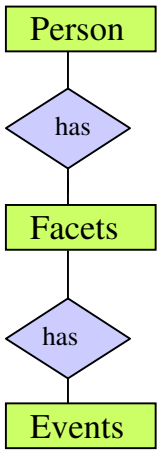


Figure 3: ERD for Schema 2 - Events and facets

Schema 3: Events, facets, aggregates, and links

Often groups of events are closely related and users would like to see them as an aggregate or provide information about all the events in an aggregate. For example, Leonardo's painting of St. Anne can be seen as an aggregate of several events (sketches, drafts, studies of faces, etc.). A physician might order multiple medications for a single diagnosis, or a college experience could be seen as an aggregate of 8 semesters, and each semester an aggregate of 5 courses. A property of aggregates is that they are likely to be summarized and presented as a single event, for example, the many steps in making the painting of St. Anne. All courses taken by a student can be summarized as a college experience. A series of various antibiotic prescriptions at different dosage might be summarized and viewed as a single aggregate.

Within the ERD we create an aggregate (Figure 4) which combines several events or aggregates. Aggregated events are closely chronological and must reside within a single facet. Aggregates are meant to represent tight clusters of events and intervals whose span of time is limited. An event or interval may participate in, at most, one aggregate.

Within the relational schema, we can define aggregates by adding a unique identification number (ID) to each event and an aggregate relation with a unique identifier for the aggregate and ID of one of the components. If there are 40 components to an aggregate, then there will be 40 tuples in the aggregate's relation.

Person (NAME, attrP1, attrP2,...)
A-events (ID, START-DATE, STOP-DATE, FACET, attrA1, attrA2,...)
B-events (ID, START-DATE, STOP-DATE, FACET, attrB1, attrB2,...)
C-events (ID, START-DATE, STOP-DATE, FACET, attrC1, attrC2,...)
Aggregates (AGGREGATE-NUM, AGGREGATE-TYPE, ID, FACET)

Tightly clustered hierarchies of sequential or parallel aggregates within a facet accommodate many situations, but there is a need for an additional mechanism to represent looser aggregations of events and intervals. A series of events or intervals may have a meaningful relationship even if they are widely separated in time or occur across several facets. A painting is generally executed over a limited period but may have used elements of studies done years earlier (i.e. separated in time) or directly inspired by a memorable travel or life-event such as the death of a relative (i.e. across facets).

Our intention is for links to not be visible normally, but when users select to view a link, they will be able to see lines or highlight the link components. Within the relational schema, we define links by an identification number (LINK-NUM), a link type (LINK-TYPE), and an ID. Each link is represented by a single tuple.

Person (NAME, attrP1, attrP2,...)
A-events (ID, START-DATE, STOP-DATE, FACET, attrA1, attrA2,...)
B-events (ID, START-DATE, STOP-DATE, FACET, attrB1, attrB2,...)
C-events (ID, START-DATE, STOP-DATE, FACET, attrC1, attrC2,...)
Aggregates (AGGREGATE-NUM, AGGREGATE-TYPE, ID, FACET)
Links (LINK-NUM, LINK-TYPE, ID)

An event or interval can participate in many link groups. Those links specified in the data are explicit links, but we envision the need for implicit links generated at runtime (e.g. by searching all events sharing a common attribute value: e.g. “hadinfluencedon=St Anne” or “physician name= Jones”).

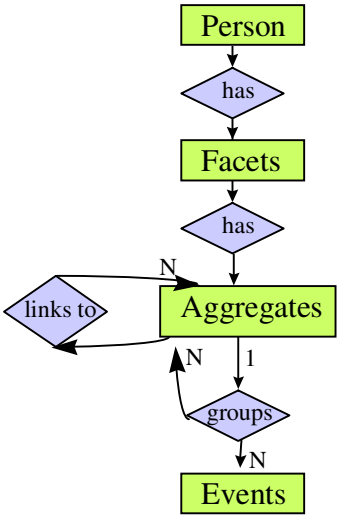


Figure 4: Final schema: Events, facets, aggregates, and links

The organization of events in hierarchies of facets is a convenience dictated by user preferences, but the definition of aggregates, the type of events they can group and the attributes of events, is dictated by real data and processes.

Medical application example

We now turn to the example of medical records. An imaging test ordered by a physician is an aggregate of one or more imaging orders (e.g. a shoulder x-ray + an arm x-ray) and one or more imaging results (e.g. the two orders may be merged to produce a single x-ray of both arm and shoulder) reviewed by the physician. This aggregate can span several days or weeks but can be summarized after the fact as a single event on the day of the x-ray, giving the outcome of the test. Here again, aggregates group tightly related events in a single facet.

On the other hand, a single visit can result in a variety of medical events including a physician note, lab tests, x-rays, drugs, and physical therapy. In this case, links need to be used to show the relationship between the events because they span several facets. The link TYPE might be “note” since all events share a common initiating note. Those links can be pre-computed and saved as tables in the data, or generated on the fly as the result of a query on all events sharing the same Note ID as the event selected by user.

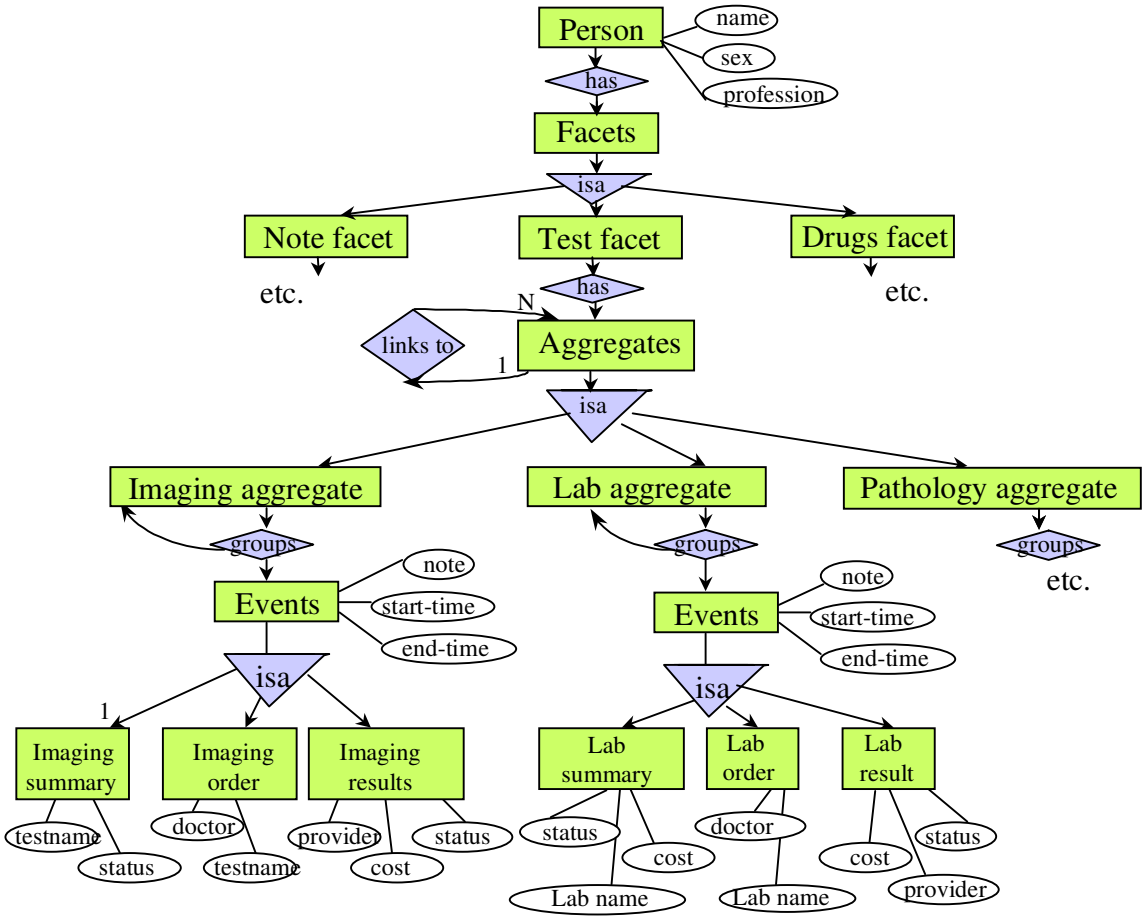


Figure 5: Partial example for medical application domain - Events, facets, aggregates, and links

From personal history data to display data

The data we described in the previous section gives semantic information about a person's life. Before users can look at this information, a conversion has to take place to map the event types and attributes to display attributes. If the display technique was chosen to be multiple textual tables, the application manager has to decide the order of the tables, the attributes to show, their order, whether to use color or font size coding or not, etc. [Isakowitz, 1995]. If a more visual display technique has been chosen, more options are available to designers to present the information (e.g. color, background, thickness, icons, animations, highlighting, etc.).

Application managers should be able to set the default values of display options for large categories of users, and individual users can adjust those values and save their personal preferences.

Figure six shows the proposed system architecture: a preference module reads the personal history data and the preference settings. It gives access to control panels allowing users to modify the presentation parameters of the personal history data.

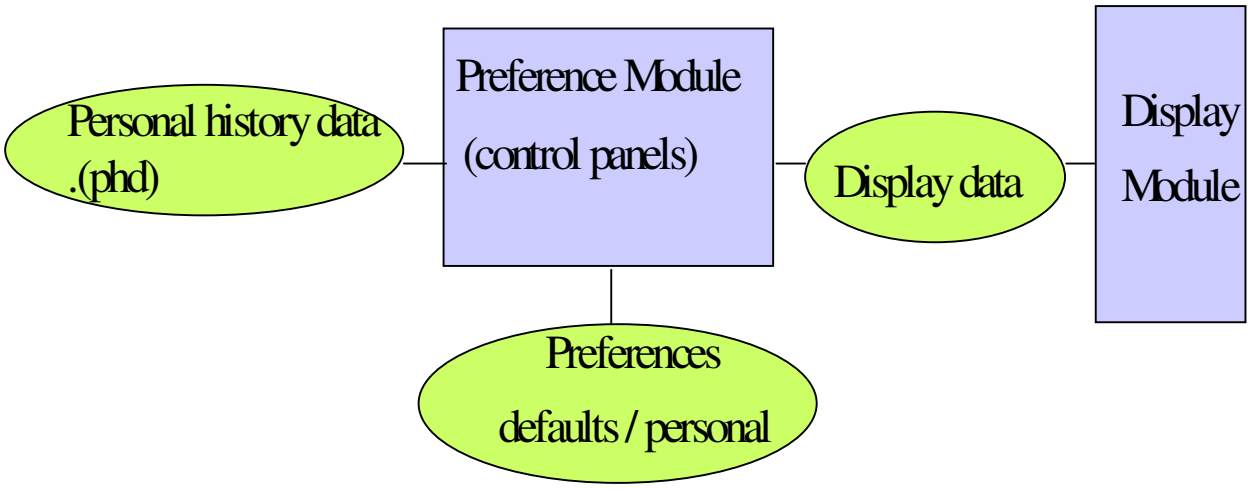


Figure 6: System architecture

The case of LifeLines

We developed a new user interface called LifeLines to visualize personal history records of delinquent youths [Plaisant, et al., 1996; Milash et al., 1996] and later applied LifeLines to medical records [Rose and Plaisant, 1996]. LifeLines provides a general visualization environment for personal histories. Its one-screen overview using timelines provides direct access to the data (Figure7). For a patient record, medical problems, hospitalization, and medications can be represented as horizontal lines, while icons represent discrete events such as physician consultations, progress notes or tests. Line color and thickness can illustrate relationships. Rescaling tools and filters allow users to focus on part of the information, revealing more details.

An experiment was conducted to study the benefits of such a graphical interface [Lindwarm et al., 1998]. Thirty-six participants used a static version of either LifeLines or a Tabular representation to answer questions about a database of personal history information. Results suggest that overall the LifeLines representation led to much faster response times, primarily for questions that involved interval comparisons and making intercategorical connections. A "first impression" test showed that LifeLines can reduce some of the biases of the tabular record summary. A post-experimental memory test led to significantly ($p<.004$) higher recall with LifeLines. Finally, simple interaction techniques were proposed to augment LifeLines' ability to deal better with precise dates, attribute coding, and overlaps.

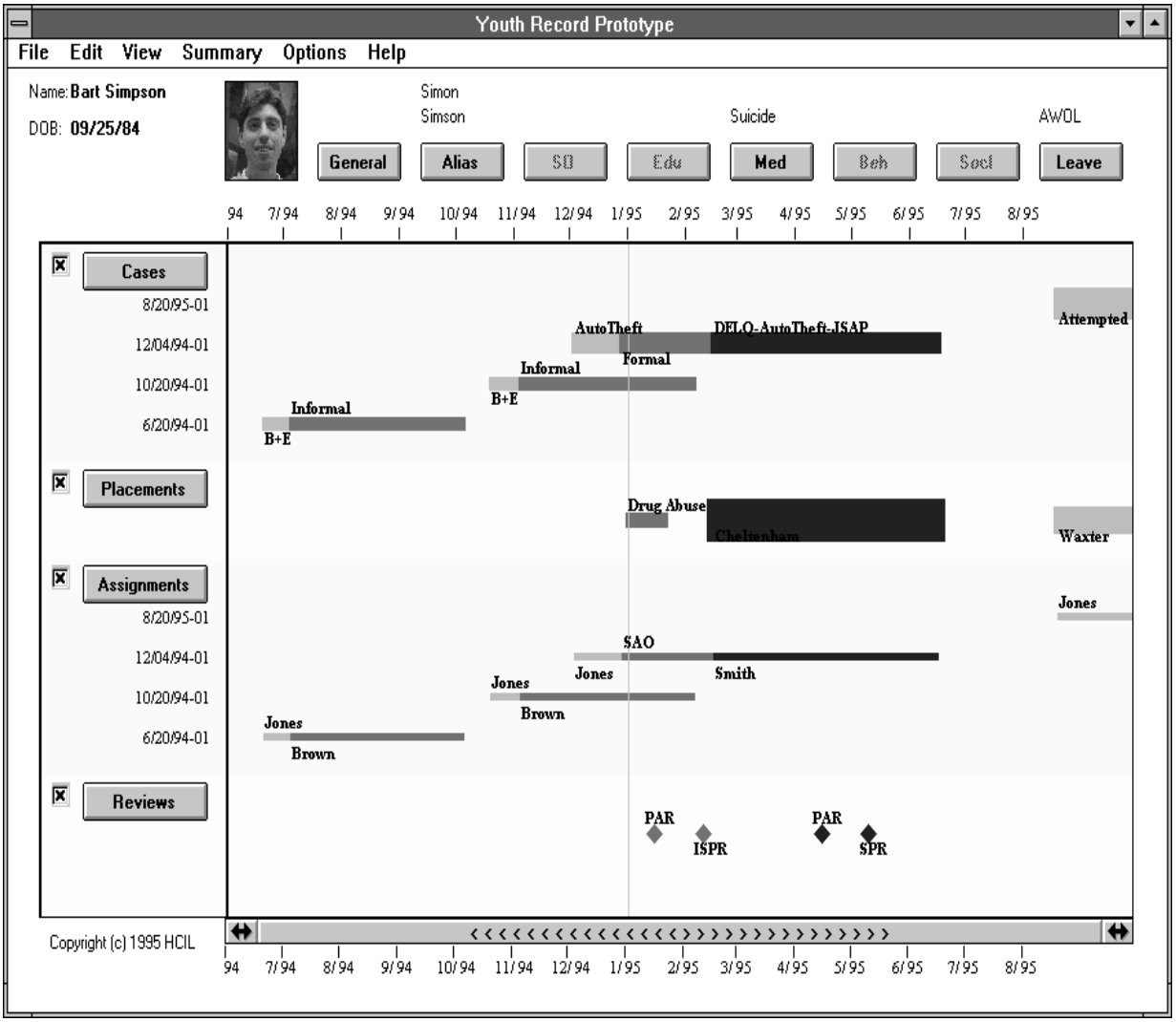


Figure 7 : Example of Lifelines for the display of Juvenile Justice youth record. The facets are Cases, Placements, Assignments and Reviews. The sequential periods corresponding to evolving case status (intake, court, adjudication, disposition, etc.) are grouped in aggregates. Placements, assignments, and reviews are linked to cases allowing users to highlight all the elements related to the case they select. The LifeLines can be zoomed to reveal more details, and acts as a giant menu as each element of the display leads to the corresponding document(s). Line thickness indicates the severity of the offense, color the depth of penetration in the juvenile justice system.

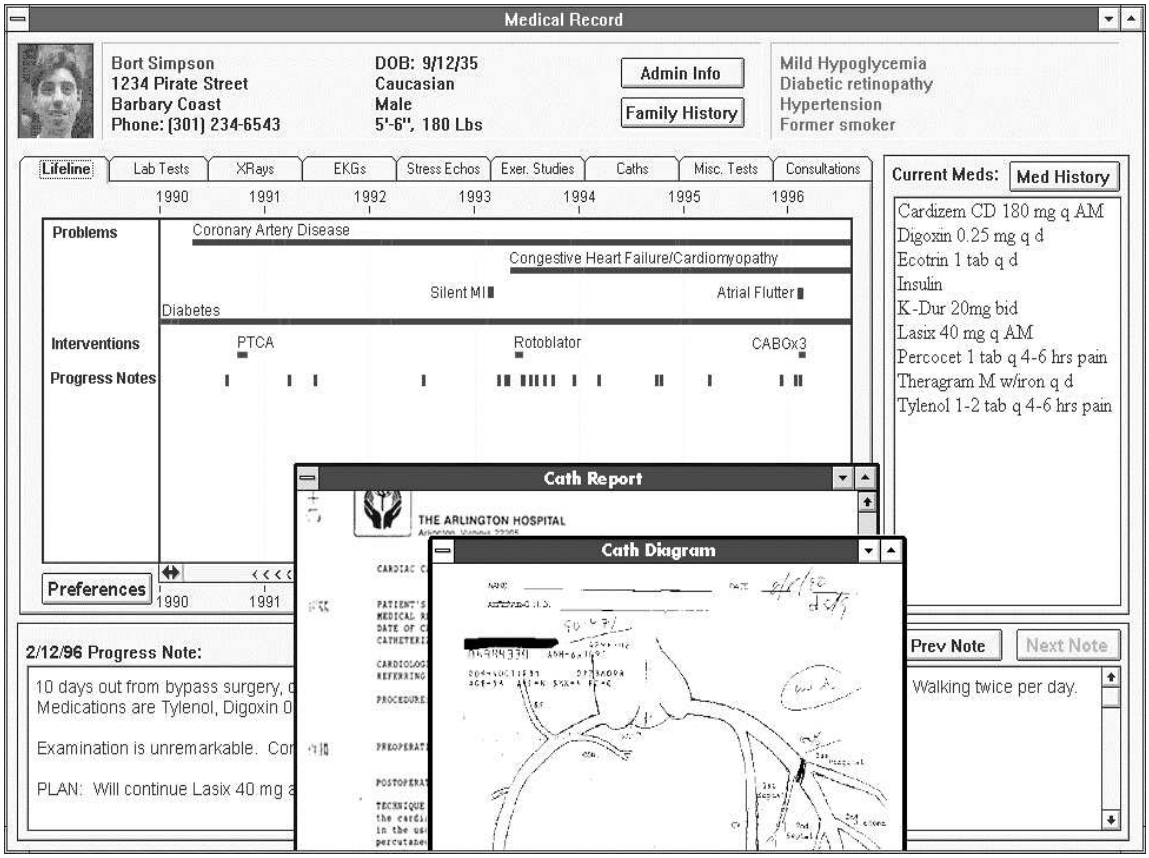


Figure 8: In this example of LifeLines for a cardiology patient record the facets chosen for initial display are problems, interventions and progress notes, medication and lab test facets can also be displayed on the LifeLines when needed.

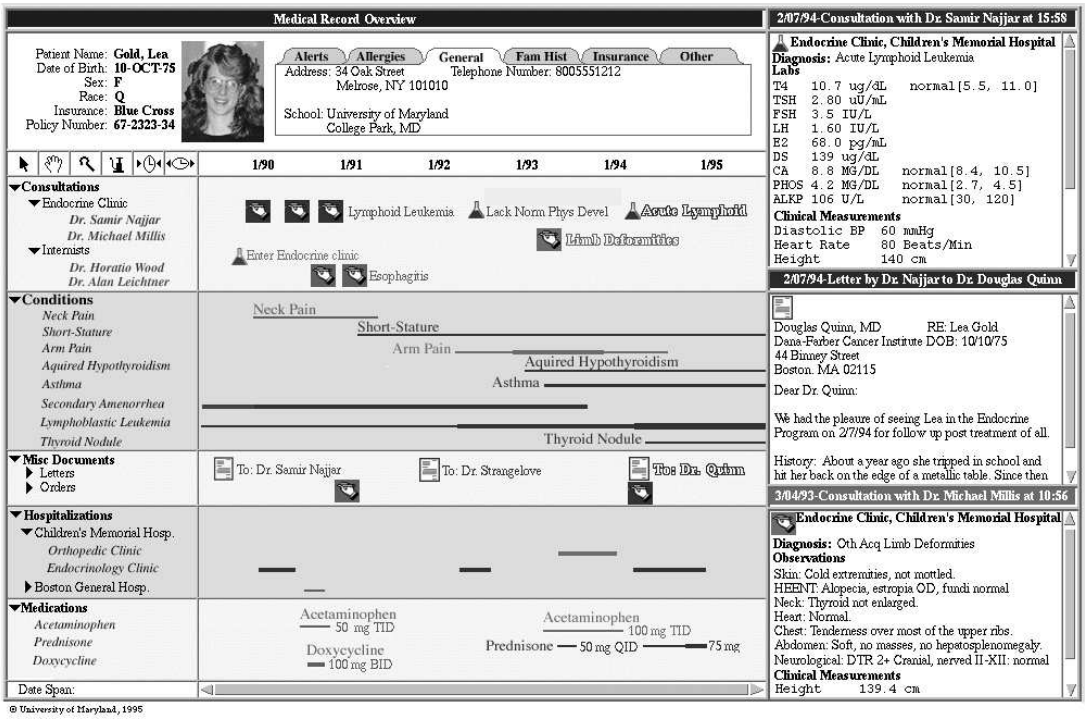


Figure 9: This alternative design (and different record) shows different facets, uses icons to show event type, line thickness to show severity of conditions or dosage of drugs. Color coding is by physician. The successive related diagnoses are aggregated to form lines of varying thickness. Drugs are linked to ordering physicians and conditions.

Our information architecture relies on a two novel data format. We designed a personal history data structure (phd) which could be the output of a conversion program from existing databases. Alternatively, new data entry programs could create the phd data structure, that has information on facets, aggregates, links, and events. Then, the control panel in the preference module enables a database administrator to specify the visual presentation and generate the *LifeLines Data* (lld). Figure 11 shows the system architecture for LifeLines, and Figure 12, the corresponding LifeLines data model. This data model is congruent to the *Personal History Data* (phd) model. It is much simpler since the personal history display data concerned with drugs, education, or, in daVinci's case, paintings can be described as points or lines with color, thickness, and labels.

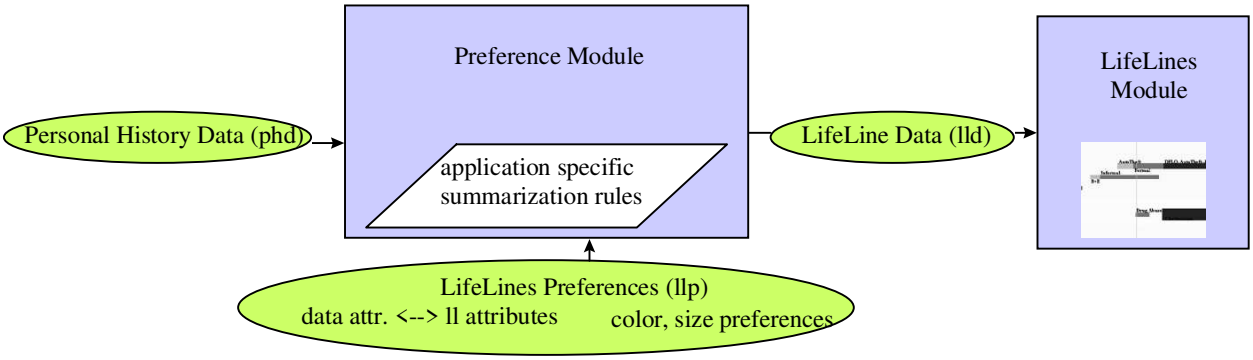
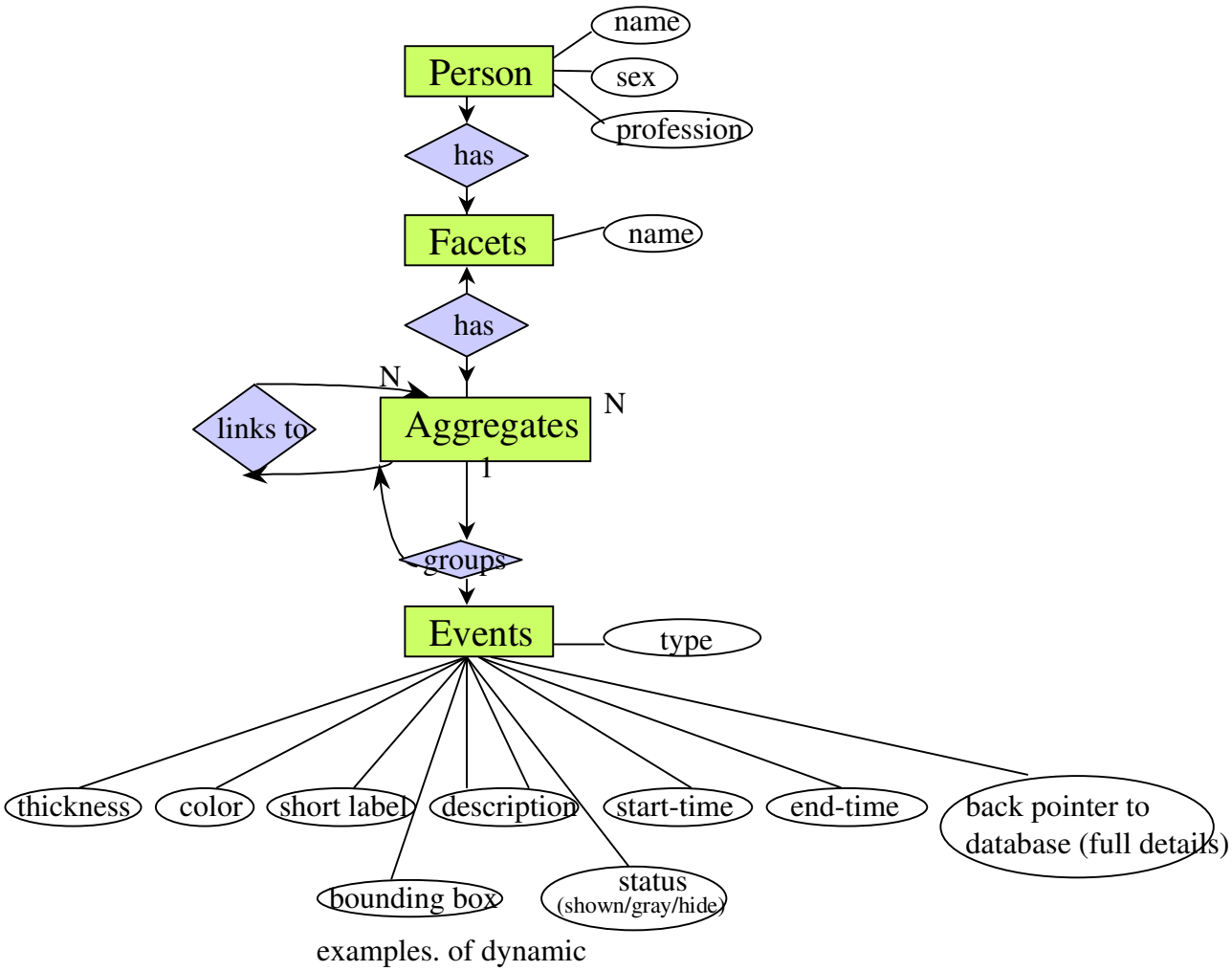


Figure 11: The system architecture supporting LifeLines. The preference data contains the mapping between Personal History Data (phd) attributes and LifeLines Data (lld) attributes. The preference module also contains the summarization rules for the data display.



examples. of dynamic

Figure 12: LifeLines Data model. The proposed attributes are the default attributes of events. Other types of low-level events can be defined with different attributes (e.g. flashing, icon files, and height from origin for plot-like display)

Summarization
Summarization is a key issue for the visualization of personal history records because of the potentially large range of life events. Since a lifetime can contain 50 million minutes, an effective visualization must deal with emergency room events as well as global overviews.

Generic summarization rules can be proposed (e.g. any 'severe" or "red' event should leave a red mark on its summary) but in the examples we studied, the summarization rules were application-specific and could only be specified by domain experts.

For example a drug in a patient record can correspond to a complex aggregate event: ordering of drug A, acknowledgment by pharmacy, dispensing of drug B [e.g. generic], re-dispensing, reordering, adverse reaction, etc. The summarization rule can also be complex: e.g.: if event type=drug, use as the summary label the label of the 1st event whose status is "dispensed" or "re-dispensed". If color is assigned to status: use the color of the last event unless there is an event whose status is "adverse reaction".

Similarly a series of orders of the same antibiotic A can be summarized as one event labeled "Antibiotic A", and a series of orders of antibiotics A, B and C can be summarized as an event labeled "Antibiotic".

Application designers may have to work with users and managers to define policies for summarization. For example: if a youth is acquitted, should the case be shown on the record summary? If it is, should the label used be the alleged offense or the word "acquitted". Those decisions can introduce bias in the interface and our experience with the Juvenile Justice system suggests that the issues will be actively discussed among users. On the other hand those summarization rules are not necessarily specific to the display technique and may have already defined to generate summaries tables, reports or statistics.

Eventually some of those rules could be specified with a rule editor and allow application programmers and advanced users to specify complex rules for data summarization. In the meantime, the application domain dependent code should be regrouped in a section of the preference module.

Proposed format for LifeLines exchange files.
Both the phd data and the lld data will most likely only exist in database format, or as runtime data structures; but to facilitate our own development (i.e. to develop the LifeLines module independently from the Preference Module) and to allow the rapid prototyping with data from other applications, we developed a simple data format to describe the LifeLines data. Our requirements were that it should be readable enough to allow someone to write simple test data files by hand, and flexible enough to allow changes as our project evolves.

```
%c comments
%person, name, sex, age, picturefile
%facet, title, bckcolor
```

```
%agg, type, #items, has-summary?
  %event, startdate, enddate, color, thickness, label, URL,
additional description
  %event, startdate, enddate, color, thickness, label, URL,
additional description
  etc.
```

Format used in the sample lld file below

```
%person, Homer Simpson, M, 40, face.gif

%facet, Notes, darkgreen

%c This note is an aggregate of all the continous diabetes
diagnosis events
%c starting with a summary event.  It also includes the
record of a canceled visit %c that will appear on the line.
The summary event always come first in the
%c aggregate.

%agg, normal, 4, yes
  %event, 1997-01-22, 1997-07-07, darkblue, thin, serious
diabetes,
  http://...,"physician=White office=greenbelt NoteID=1"
  %c note that the additional description contains attributes
that have no visual
  %c elements but can be searched or displayed in a separate
table.
  %event, 1997-01-22,1997-01-30, lightblue, p2 , light
diabetes,,"NoteID=1"
  %event, 1997-01-30,1997-05-03, mediumblue, p2, medium
diabetes,,"NoteID=1"
  %event, 1997-05-03,,black, p4, cancelled visit,,"NoteID=1"
  %event, 1997-05-03,1997-07-07, darkblue, p6, serious
diabetes,,” NoteID=1"

%c Another simpler note:
%agg, normal, 1, no
  %event, 1990-01-01, today, lightblue, p4, obesity,,"
physician=Rose,
  office=greenbelt,  NoteID=2"


%facet, Imaging, lightgreen

%c  This aggregate groups together a battery of 4 tests (3
have been reviewed already, one is pending)
%agg, normal,4,no
  %event, 1997-08-05, 1997-08-10, black, p2, normal test
A,,"NoteID=1"
  %event, 1997-08-05, 1997-08-10, black, p2, normal test
B,,"NoteID=1"
```

```
  %event, 1997-08-05, 1997-08-20, orange, p2, see note of
test C,
  http://testcnote.html,"NoteID=1"
  %event, 1997-08-05, today, magenta, p2, testD result
pending review,,"NoteID=1"

%c This imaging shows nested aggregates.  The physician
ordered 2 xrays for arm and %c shoulder but the provider
combined the orders and returned a single result .  This %c
kind of detail is useful for recent tests but may not be
included in the lld of %c older records.
  %agg, normal,2,yes
%summary, 1997-01-15,,black, p2, xrays,,"provider=Carl&Smith,
cost=$298, NoteID=1"
  %agg, normal,2,no
%event, 1997-01-05, 1997-01-15, black, p2, ordered shoulder
ray,,
%event, 1997-01-05, 1997-01-15, black, p2, ordered arm ray,,
   %agg, normal, 1, no
     %event, 1997-01-15, 1997-01-18, black, p2, normal
results shoulder+arm Xray,,
```

Sample LifeLines data (lld) file containing two facets: Notes and Imaging

The display of this sample data file is shown in Figure 13a and b.

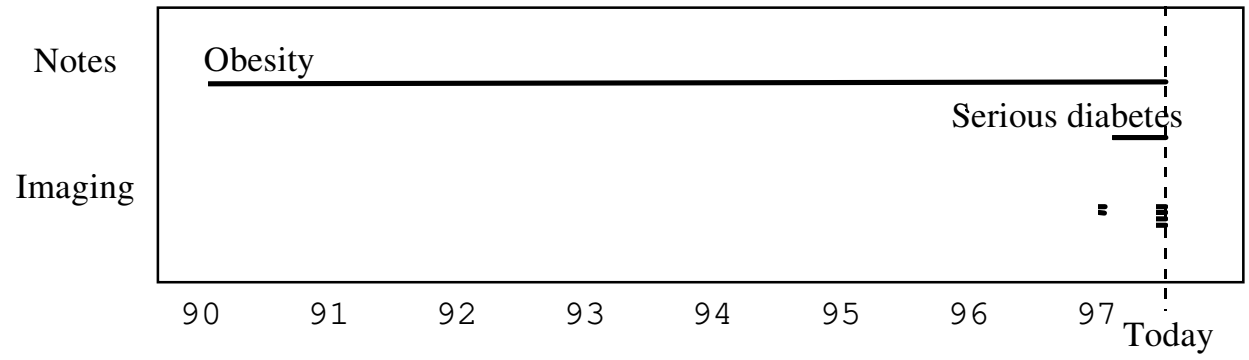


Figure 13a : The record overview shows the entire record. If space is tight, the summary information is used (e.g. here for the diabetes.)

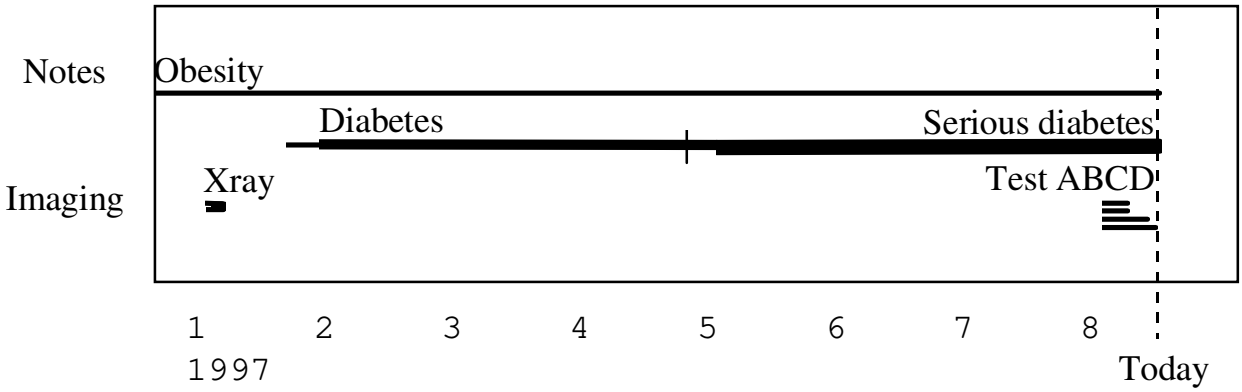


Figure 13b : When the user zooms on recent months (either with sliders, by clicking on "today" or by selecting and focusing on the Diabetes line) more details appear. The 3 stages of the diabetes are visible, and the recent test shows that 3 test results are back (color would give the abnormal vs. unremarkable status) while the 4th is still pending (i.e. extends to today and color indicates pending status). A click+focus on the xray would zoom again and show the details of the multiple order + result aggregate.

User control of LifeLines display attributes

The preference module will play an important role in the adaptation of the display to the user task. In harmony with general user requirements the application manager for a hospital might set different default settings for physicians and administrators. For example administrators do not see the notes facet but the insurance payments facet instead. Different specialties will prioritize facets differently (e.g. the immunization facet will probably be most useful for pediatricians). But individual users will also need to modify the display parameters as their need varies. Color blind users may need to switch to colors or shades they can differentiate. The line thickness might be associated to severity or associated cost, color assigned to physician or outcome. Many options are available to control the interactive display. The label of the event shown when the cursor passes over the event can be displayed in a temporary overlapping window or in a dedicated space below the LifeLines. Some users will prefer to use double clicking to zoom on an event instead of jumping to another screen showing all detail about the event. Some users might prefer to show links by drawing lines between linked events, instead of having non-linked events grayed out. Finally, summarizing rules may be customizable: some users may prefer to ignore unremarkable events during the summarization while others may want to keep that information too.

We categorize the type of preference settings as follows:

Preference settings that can be specified using simple control panels and are application domain independent:

- Selection and ordering of facets
- Mapping between personal history attributes and display attribute, and mapping of attribute values (a sample control panel mockup is shown in Figure 14)
- Mapping of users' actions to display actions (e.g. select, focus timeline, show menu of linked events, show all attributes, jump to complete details)

- Simple window management (size, position and aspect ratio of LifeLines window)

Preferences that require specialized tools and application dependent codes:

- Summarizing preferences (might require a complex rule editing module, or application domain dependent code)
- Advanced window management preferences (coupling among multiple LifeLines windows (overview/ detail timeline), placement and coupling of LifeLines vs. full detail document windows. Multiple opening of documents related to a single event, etc.)

We have developed a Java prototype of the LifeLines module. The phd data is read from a relational database (Microsoft Access) using a data model similar to the one described in Figure 5. A control panel written in Java allows users to select facets and map phd attributes to display attributes, and generates the lld data for the LifeLines applet.

	FACET SELECTION/ORDERING	ATTRIBUTE MAPPING	
Show	**Facet and event types**	**Event attributes of**:	**LifeLines attributes**
X	Notes	Imaging result	
x	Notes	test name	label
x	Visit	test code	
x	Tests	review	color
x	Imaging order	cost	thickness
x	Imaging result	provider	
x	Lab order	insurance	[ignore]
x	Labs result	nuclear exposure	[ignore]
	Immunization		
x	Medications		
x	drug order		
x	drug dispense		
x	adverse reaction		
	etc.		
		ATTRIBUTE VALUE MAPPING	
	[selected facet can move up and down with arrows]	**Values of** : test name	**Values of:** label use as-is allow truncating : yes/no
		Values of : review	**Values of:** color
		unremarkable	black
		see note	orange
		abnormal	red
		pending	purple
		Values of : cost	**Values of:** Thickness
		0-200	(pixels)
		201-500	2
		500-2000	4
		2000+	6
			8

Figure 14: Example control panel allowing users of application managers to modify the display parameters. Here the user chose not to show the immunization facet and is currently mapping the attributes of imaging results. Because the number of attributes can become very large users can also mark some attributes with an "ignore" flag. The corresponding data will not be available at runtime for querying or remapping. (e.g. "insurance" will not be in the lld data, while "provider" remains available for rapid querying at runtime).

Conclusion

Records of personal histories are needed in a variety of applications. Members of the medical and legal professions examine a record to garner information that will allow them to make an informed decision regarding their patient or case. Decision-making critically depends on gleaning the complete story, spotting trends, noting critical incidents or cause-effect relationships, and reviewing previous actions. Professional histories such as résumés help employers relate a prospect's skills and experiences to employment and education. Financial and retirement plans associate past and upcoming events to culminate in an expected result. In most applications, delays in gathering the information to construct a meaningful overview of the record can have deleterious effects. In a medical situation, a patient's treatment may be delayed while charts and lab results are assembled. In a social work situation, assistance to a youth in detention may be delayed for weeks while school and court records are brought together.

While more attention is now put on developing standards for gathering and exchanging personal records (especially in the medical field), little effort had been applied to designing appropriate visualization and navigation techniques to present and explore personal history records.

This paper proposes an information architecture for personal history data and describes how the data model can be extended to a runtime model for an intuitive visualization using graphical timelines. Our information architecture, is developed for medical patient records, but is usable in other application domains such as juvenile justice or personal resumes. Future challenges include better algorithms to rapidly display and animate large records; visual rule editors for user control of the summarization process; and distributed architectures to gather data from multiple sources.

Acknowledgments

This information architecture greatly benefited from the long discussions with Rich Mushlin, of IBM T. J. Watson Research Labs, who used his knowledge of the medical patient record data architecture to challenge our proposed architecture. Dan Heller and Jia Li also contributed to the discussion as they implemented the phd data in MS Access, the control panel, and LifeLines interface in Java. We also want to thank Christos Faloutsos for his advice on ERM. We greatly appreciate the support provided by IBM through its Shared University Research program, and NSF grants.

References

Allen, R.B., Interactive timelines as information systems interfaces, *Symposium on Digital Libraries*. Japan, August 1995.

Allen J.F., Maintaining knowledge about temporal intervals, *Communications of the ACM 24,* 11 (November 1983), 832-845.

Hibino, S., Rundensteiner, E. A., User interface evaluation of a direct manipulation temporal visual query language, *Proc. ACM Multimedia 97 Conference*, ACM, New York (1997), 99-107.

Isakowitz, T., Stohr, E., and Balasubramanian, P., RMM: Methodology for hypermedia design,
Communications of the ACM 38, 8 (August 1997), 34-44.

Karam, G.M., Visualization using timelines, *Proc. of Intl. Symposium on Software Testing and Analysis (ISSTA),* available in *SIGSOFT, ACM Software Engineering Notes,* ACM, New York (1994).

Kilman, D. and Forslund, D., An international collaboratory based on virtual patient records, *Communications of the ACM 40*, 8 (August 1997), 111-117.

Kumar, V. and Furuta, R., Modeling historical information, Technical Report Texas A&M University, Department of Computer Science, College State, TX (1997).

Lindwarm D., Rose, A., Plaisant, C., and Norman, K., Viewing personal history records: A comparison of tabular format and graphical presentation using LifeLines, *Behaviour & Information Technology* (to appear, 1998).

Mackinlay, J.D., Robertson, G.G., and Card, S.K., The Perspective Wall: Detail and context smoothly integrated, *Proc. CHI '91 Conference*, ACM, New York (1991), 173-179.

Milash, B., Plaisant, C., and Rose, A., Exploring LifeLines to visualize patient records, video in *CHI '96 video program* (Vancouver, BC, Canada, April 13-18, 1996), ACM, New York.

Plaisant, C. and Rose, A., Exploring LifeLines to visualize patient records ,University of Maryland CS-TR-3620, CAR-TR-819 (March 1996). A short version of this report appeared as a poster summary in *1996 American Medical Informatic Association Annual Fall Symposium* (Washington, DC, Oct. 26-30, 1996), 884, AMIA, Bethesda MD.

Plaisant, C., Milash, B., Rose, A., Widoff, S., and Shneiderman, B., LifeLines: Visualizing personal histories, *Proc. CHI '96 Conference* (1996), 221-227, color plate 518.

Sanderson, P., Scott, J., Johnston, T., Mainzer, J., Watanabe, L., and James, J., MacSHAPA and the enterprise of exploring sequential data analysis (ESDA), *Int. J Man-Machine Studies 41* (1992), 633-681.