User Interaction in a Sketch-Based GIS User Interface*

Andreas D. Blaser
National Center for Geographic Information and Analysis
and
Department of Spatial Information Science and Engineering
University of Maine
Orono, ME 04469-5711, USA
abl@spatial.maine.edu
http://www.spatial.maine.edu/~abl/abl.html

Abstract

The art of sketching and drawing has been familiar to humankind for millenniums and beside spoken language and gestures the graphical representation of information has been an important method to communicate knowledge and experience. In human-computer interaction of today's geographic information systems the natural skill of sketching plays an ancillary role, because most user tasks are strongly tied to an interaction based on typing supported by primitive forms of interaction, such as pointing for selection. Although this approach works fine for a host of applications, it can become difficult to formulate even simple statements as soon as spatial, hierarchical, or conceptual structures are to be communicated. A typical field of application where such problems frequently arise is querying a spatial database, because traditional query methods—which imply text-oriented query languages—lack the power and expressiveness necessary to describe complex spatial configurations. This situation can be improved through the introduction of new user modalities, such as *Computer Interpreted Sketching*. This paper analyzes traditional and alternative user interaction methods and explores the potential of a sketch-based user interface with the focus on applications in GIS. It is argued that sketching and drawing gestures are well suited for querying a spatial database and that they can be efficiently applied for browsing and updating such a database.

1. Introduction

Sketching, particularly freehand sketching, was known by almost all ancient tribes and people who wrote their own history. Neanderthal men sketched their everyday lives on cavern walls, Pythagoras drew his circles into the sands of Greece's beaches in order to prove his mathematical theories, and still today, sketching is considered a very important skill. Long before people learn to write, they learn to sketch. Children at the age of three to four are able to make a drawing of a situation that is much richer than their verbal explanation of the same situation without a sketch (Piaget and Inhelder 1967). A sketch can be made to explain something or to raise a question. The elements in a sketch can be physical as well as non-material. The same freedom of expression applies to the relationship among drawn objects: a relationship can be, for example, established simply through a specific composition of objects or through the choice of common object characteristics, such as shape, color, or orientation.

Unfortunately these inherited powerful expression mechanisms in sketches are not yet integrated in the human-computer interaction of today's commercially available GIS or database applications (Egenhofer and Herring 1993). Standard human-computer dialogs that use verbal forms of expression are often not meaningful enough to handle non-standard descriptions of spatial situations. Hence, formulating a spatial database request that goes beyond a common spatial query becomes immediately very complicated and error prone.

* This work was partially supported by Rome Laboratories under grant number F 30 602-95-I-0042.

People are well trained in using and understanding spatial predicates, such as “across“, “along“, or “close”, whereas computers—by their very nature—are optimized for precise and unambiguous input and operations with values in tables, arrays, or lists. In order to overcome this gap between the way a user naturally expresses ideas, queries, or intentions and the way a computer can satisfactory interpret them, we need powerful user interfaces (Mark and Gould 1991). Current systems leave a user the choice between formulating a spatial query by typing a query statement with one of the current spatial query dialects (Egenhofer 1992; Egenhofer 1994), and composing the same query by choosing a sequence of commands from pull-down menus or pre-formatted icons in a graphical user interface. This concept of user-computer interaction releases the user only from remembering the particular query syntax, but not from translating his or her mental model of the query into a synthetic, application-centered form.

Neither of these approaches is intuitive and we propose an alternative user interface, in which a user can formulate queries by sketching them. The concept of this user interface and the associated query language is called *Spatial-Query-by-Sketch* (Egenhofer 1996b). Spatial-Query-by-Sketch attempts to overcome limitations of conventional spatial query languages by the use of advanced technologies, such as pen user interfaces, and powerful reasoning mechanisms that are based on a comprehensive theory of spatial relations. Sketching is an easy and meaningful form of expression, that also will allow non-specialists and casual GIS users to successfully query and operate a spatial database. Such a simple operation is of great importance, because GIS applications have found their way into a host of new application areas, such as utilities or emergency services, increasing the number and the diversity of potential GIS users even more. A powerful system must, therefore, satisfy users without education in computer-sciences or geo-sciences as well as GIS professionals (Medyckyj-Scott and Hearnshaw 1993).

The remainder of this paper continues with a short review of user interfaces that have used sketching for input generation and our vision of a sketch-based device. Sections 3, 4, and 5 discuss possible user modalities, user actions, and user operations in a sketching environment respectively. The paper closes with a summary of the findings and some perspectives for future work concerning sketch-based user interfaces.

2. Sketch-based User Interfaces

Sutherland’s *Sketchpad* (Sutherland 1963) was the first sketching application on a computer and a cornerstone of applied computer graphics. Sketchpad’s primarily concern was the construction, composition, and replication of geometrical objects. For this purpose Sketchpad used such advanced devices as a pen for direct drawing on a CRT monitor. This is even more impressive, because at that time punch cards, line printers, and ASCII terminal were standard I/O devices. An other pioneering application in this domain was *ThingLab* (Borning 1979; Borning 1986), which consists of a library of SmallTalk modules as the base for constraint formulations among objects. More recently, Gross introduced the *Electronic Cocktail Napkin*, a pen-based drawing environment for early design stages (Gross 1994; Citrin and Gross 1996; Gross 1996), which incorporates constraint, shape, and character recognition features.

Other applications that involved sketching, such as *Penguins* (Chok and Marriott 1996), *Minnie* (Spence 1974; Spence, Cheung et al. 1986), and *Image Retrieval by Sketch* (Del Bimbo and Pala 1995), were built on the basic principles of Sketchpad, adding more functionality or application-specific features into their interfaces. Sketching was also employed as an alternative input source for image and painting retrieval applications such as *Query by Visual Example* (Kato, Kurita et al. 1992).

Sketching in computer science is either used for construction or retrieval purposes. Applications for constructing tend to transform a sketch directly into a constraint- or rule-based geometric representations as soon as drawn objects are detected. Objects are converted into symbols from a template and sequences of characters become ASCII strings. Detected links between objects are stored and this relation is kept when the sketch is edited. Certain applications can keep the initially sketched objects on the drawing surface and do this conversion in the background, such as the Electronic Cocktail Napkin. Minnie, the electronic circuit designer's tool, substitutes drawn objects immediately with the corresponding symbol. This method of matching drawn objects with templates is very efficient for application fields with a limited set of possible objects—a flowchart is a good example—but as soon as there is a multitude of potential entities with sub-forms and many specialties, it becomes very difficult to be complete without taking the local or global context into account.

Applications that are concerned about image retrieval can be divided into two classes: The host of implementations applies techniques that seek to match the shape or the outline of drawn figures with image extracts from the database of the same type. These applications do not detect objects and do not take context into account. Query by Visual Example is an example. The sketch needs, therefore, not to be changed into a “correct” geometrical form. The other class of applications is more content-oriented: drawn objects are classified and, depending on the context, they are given a refined meaning. This classification implies in some cases also that the drawn object is changed into a symbolic representation (Citrin and Gross 1996).

These different approaches that employ various techniques have focused primarily on the application itself, but so far little attention has been paid in taking advantage of the direct interaction that a pen-based user interface has to offer. An other exception is the Electronic Cocktail Napkin (Gross 1994), which provides users with pen-based gestural commands, such as draw, erase, copy, undo, or pick (Figure 1). These gestures can be trained and recognized much like handwritten characters.

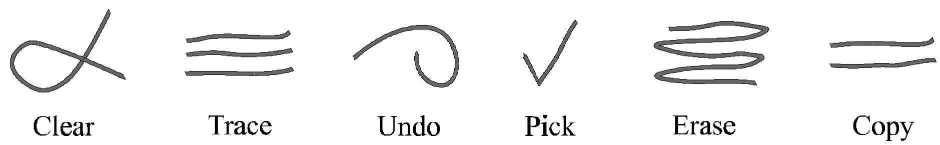


Figure 1 The set of gestural commands provided by the Electronic Cocktail Napkin.

Similar to the Electronic Cocktail Napkin we associate in Spatial-Query-by-Sketch the term of s*ketching* with freehand drawing rather than with the construction of geometric figures. By choosing this approach, we try to come as close as possible to the original meaning of the electronic paper metaphor. Spatial-Query-by-Sketch is complementary to and builds on existing techniques known in user interfaces, because it can involve both keyboard and pen input. Depending on knowledge, skill, and liking, multiple expression mechanisms must be permitted—sketching, freehand writing, typing, selecting menus, and maybe also verbal interaction should be freely interchangeable. The sketch-based system should be user-centered and, therefore, it has to be scaleable and user adjustable, without being inconsistent.

The manner of interacting with a computer system is dictated by both software and hardware, therefore, we want to take a quick look at a device tailored for sketching. It is very likely that this system will be used for many other applications besides GIS, ultimately it will be a substitute for things like scratch paper, notebooks, agendas, books, and calculators. It might be the only computing device people carry around. Hence, Spatial-Query-by-Sketch will only be one of many applications running on this device.

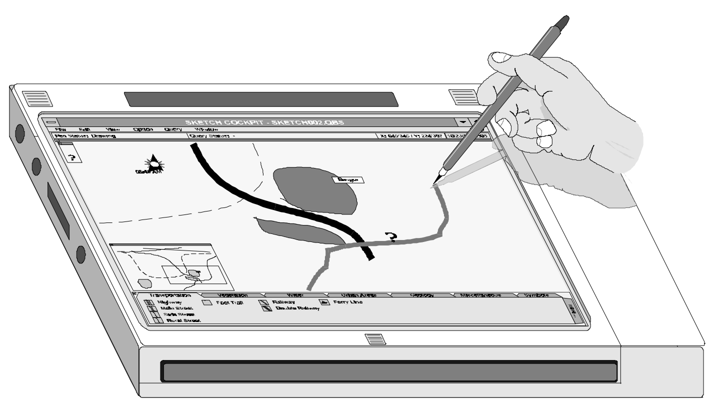


Figure 2 Example of an autonomous, multi-functional, sketch-based computer (with a retractable keyboard, multimedia capabilities, and a set of connectivity options).

Figure 2 shows our vision of a multi-functional sketch-based computer, where the pen has replaced the mouse or any other conventional input device (except the integrated retractable keyboard). A typical sketch-based computer is modular, it will be light in weight and its format is about the size of a letter legal or A4 notepad. It has no moving parts and will work either connected to a larger system or autonomously. The drawing surface is sturdy and easily interchangeable and an interaction with a finger is as well possible as the use of an ordinary pencil.

In order to understand the basics of user interaction in a sketch-based user interface we have to explore the interplay between user modalities and actions. The following sections will, therefore, identify important elements of human-computer interaction in a user interface, primary from the user's perspective and with an emphasis on sketching. The last section is dedicated to user operations and especially operations applied to objects with spatial components, such as geographic objects.

3. User Modalities

Modalities address any type of sensation, including vision, hearing, and various ways of expression, such as writing, talking, or gestures with which people interact daily with each other (Neal, Bettinger et al. 1988). A multimodal interaction involves more than one form of communication at the same time. An example of a multimodal interaction between two persons is if somebody draws a sketch and simultaneously explains this sketch verbally. A multimodal user-computer interface is accordingly defined as an interface that offers multiple concurrent input and output channels. Multimodal communication among people is quite common. In addition to speech there are less prominent human communication channels, such as eye-contact, face-mimic, and other body-signals or gestures, which help us daily to exchange information. Modalities can be used complementary or stand alone, they can support or contradict each other, in the latter case the exchange of information becomes unclear and ambiguous. The effectiveness and performance of an interaction depends on the characteristics of the chosen modalities, their synchronization and on the capabilities of the information sender and receiver. Which modalities are chosen in a specific situation depends on various factors, including the task to accomplish, personal skills, mood, and the availability of modalities (Kuhn 1992).

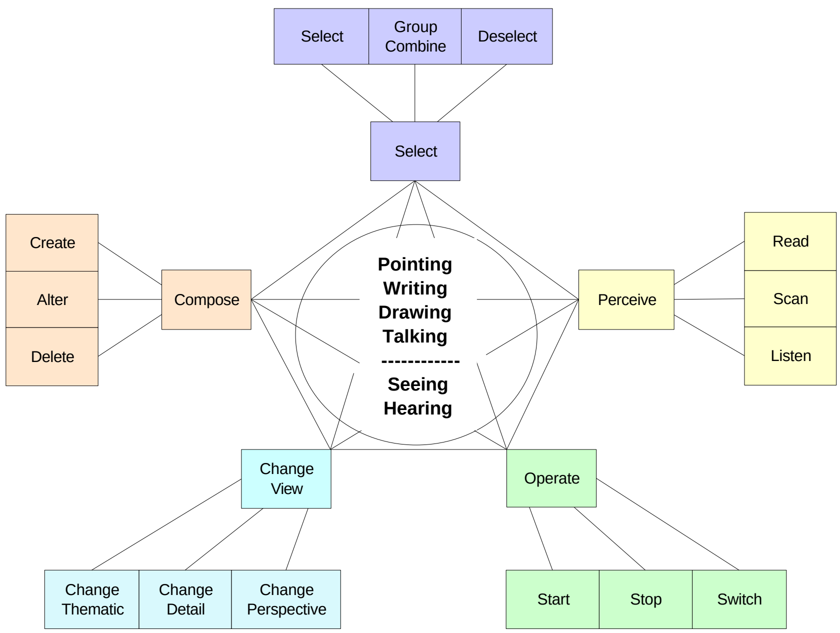


Figure 3 The basic user modalities and elementary user actions involved in human-computer interaction in a sketch-based user interface.

The interaction with a computer is quite limited compared with the rich forms of inter-human communication. Most of today's systems allow users only to point and write in order to generate input and they produce—aside from electromagnetic waves of the monitor and some beeping sounds—only text and graphics on the computer screen. This situation could be improved, if the field of possible and processable modalities would be enhanced with visual and acoustical communication techniques, such as sketching and talking. Figure 3 demonstrates the proposed set of elementary user modalities and user actions on which a human-computer interaction in a sketching user interface can be built.

4. User Actions

The term *user action* signifies an elementary user-computer interaction, which can be of an active or passive type. Active is connected with *doing* something, while a passive action is associated with *perceiving* something. We can distinguish five types of elementary user actions that do not overlap and on which any human-computer interaction can be built on: *Perceive*, *select*, *compose*, *change view*, and *operate*.

Perceive, select, and compose are primary activities, while change view and operate are supporting activities. An action almost never stands alone. In general, elementary activities are combined to composite actions, which then are called *operations*. The efficiency and usability of a system can be measured by counting the number of elementary actions that are necessary to perform specific operations. In addition to this quantitative approach, it is possible to anticipate the quality of a system by considering the simplicity of individual actions, for instance, a direct selection with a pen on a touch screen is probably more natural than an indirect selection with a mouse. It can, therefore, be derived that operations in a well balanced system require a minimum of easy and intuitive actions.

	Actions:	**Perceive**	**Select**	**Compose Text**		**Compose Drawings**		**Change View**	**Operate**
Modalities:				Simple	Complex	Simple	Complex		
Output:	Pointing	—	●●●●●	●	❍	●●●	●	●●●●●	●●●●
	Writing	—	●	●●●●●	●●●●●	●	❍	●	●
	Drawing	—	●●●	●	❍	●●●●●	●●●●●	❍	❍
	Talking	—	●●	●●●●	●●	●	❍	●●	●●●
Input:	Seeing	●●●●●	—	—	—	—	—	—	—
	Hearing	●●●	—	—	—	—	—	—	—

●●●●● excellent, ●●●● good, ●●● fair, ●● limited, ● poor, ❍ inadequate

Table 1 Hypothetical qualitative rankings for user modalities with respect to their suitability for specific user actions.

Table 1 lists the strong and weak points of the seven principal user modalities in Figure 3. It is conspicuous that the stability and flexibility of user interactions can be significantly improved if drawing and talking are included into the set of possible user modalities. Compose actions will benefit directly from these alternative modalities, while actions focusing on perception, selection, or operation are provided with a backup modality.

5. User Operations

A user operation incorporates in general several elementary actions, involving multiple modalities and various human-computer I/O cycles. User operations that need fewer actions are faster, easier, and in general more reliable, because less sources of errors are involved. On the other hand it depends on the complexity of each single action. The discussion in the previous section lets us assume that actions used in a sketch-based user interface are similar to those employed in one of today's graphical user interfaces, but that the applied modalities are different. Alternative modalities such as sketching and talking will significantly broaden the human-computer interface, leading to a richer and more dynamic interaction.

While talking can be used in many cases as a substitute for typed input or to enhance the performance of interaction—because it is fast and flexible, and because it can be used in parallel to other modalities—interpreted sketching provides yet an other dimension. Sketching lets a user easily express spatial, hierarchical, or conceptual problems, ideas, or intentions without any difficult translation of the mental model's original notion. Figure 4 focuses on three common initial starting points of a user-computer interaction and on possible resulting operations in a GIS environment.

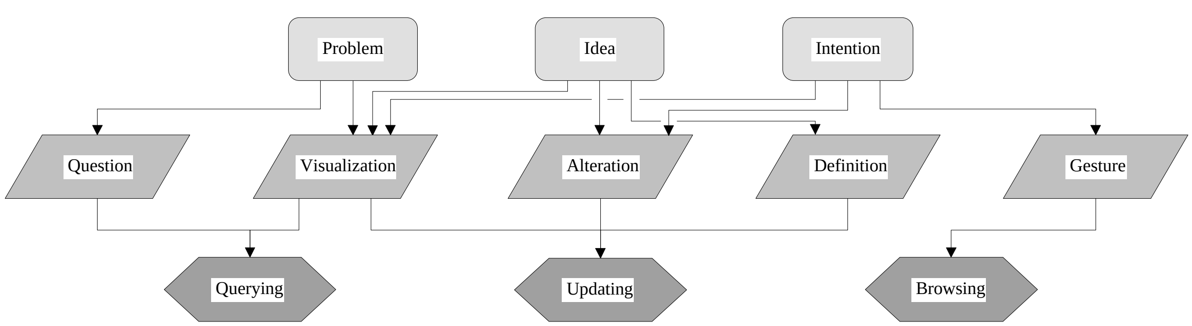


Figure 4 Initial statements, possible formulations, and the resulting user operations.

A problem, for instance, can be formulized as a question or as a visualization of relevant conditions. Both can then be transformed into a sketched query. Visualization offers yet an other possibility: Through the graphical representation of involved objects and relations, the visualization itself may contain the solution to the problem, which was not conceivable before its formulation (Larkin and Simon 1987). In this case the visualization can also directly lead to an update operation which may involve sketching as well.

According to this example and starting with either a problem, an idea, or an intention, three major user operations that can involve sketching can be distinguished: *Querying*, *updating*, and *browsing*. Sketching can be applied to all of these user operations, what embodies a great application potential because these user operations are part of the vast majority of spatial database related task within a GIS. To explore the effect that sketching can have when working with a GIS, the following sub-sections are devoted explicitly to browsing, querying, and updating.

5.1 Browsing in a Spatial Database

Thanks to the simplicity of browsing, large amounts of connected information can be conveniently examined and it is, therefore, most likely that browsing is an essential part of many future applications including GISs (Kuhn 1992). This broad application field of browsing will also have implications for the user structure, which will include users of the whole bandwidth of skill levels.

In order to allow browsing and navigating, a spatial database must have visually selectable objects that are linked among each other. Without such a connecting structure it becomes very difficult to analyze a spatial database with a visual approach. Navigation rarely leads directly and in a single step to the desired answer. It is more comparable with an iterative or recursive process of touching on the intended results, where each new question is based on the answer of the previous question, which again implies that there is a starting point, similar to a company's homepage with a choice of predefined links to related topics. The starting point is also an important difference between navigation and an explicit user query, where the user has to formulate a specific question right from the beginning.

It is easy to anticipate that direct interaction with a pen will be immediately advantageous for navigation and browsing, because it reflects people's natural temperament. Unfortunately the set of today's modalities used for this purpose is limited essentially to pointing—foremost in order to select predefined links—and formulating simple text-based queries with basic logical operators. An encounter of a situation where pointing and verbal description alone are impracticable is, therefore, very likely. This is where drawn gestures can be effectively introduced. Imagine an application that lets a user browse through text and pictorial information, and where objects of interest can be specified with a drawing gesture, such as circling. The use of gestures allows a user also to select a group of objects or to execute *simple operations*, such as an intersection or a change of view, on selected objects that have been so far retrieved (Figure 5).

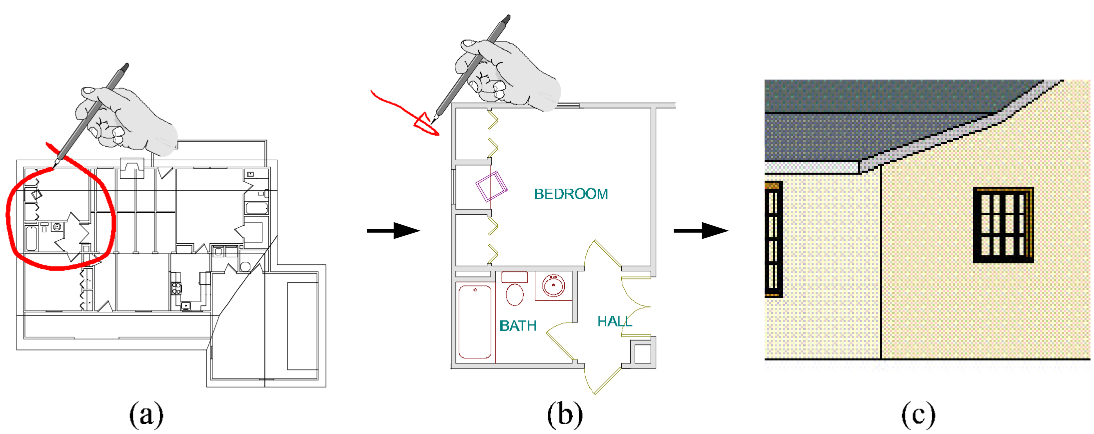


Figure 5 An application of gestures in an architectural design database. (a) a grouping selection, (b) the indicated direction of a perspective, and (c) the resulting perspective itself.

The use of gestures goes much beyond a selection by pointing or following hard coded hyperlinks, because the system takes also into account the context, in which the gesture occurred. Gestures are valuable tools for the examination of already retrieved information, such as the results of a query. Initially and at a more general level, information is preferably represented graphically by visual maps or metaphors taken from our daily environment or from a specific field of application. With reference to the example in Figure 5, the initial metaphor would be a conventional ground plan on paper, to which the architect is accustomed to. At a later point, when the search has narrowed down and at a level to which only interested users advance, the use of tables, lists, and other textual description is adequate beside graphics as well.

Despite their great potential, browsing, and navigating are limited by the extent of information that is compiled and available in the database or that was retrieved by a search engine. Much depends, therefore, on the definition of a *simple operation* and its capability to generate new views on top of the data. In our example in Figure 5, for instance, we assume that the generation of a 3-D perspective from an architectural ground plan belongs—beside standard operations such as zooming or panning—to this set of simple operations. According to the specific field of application, the set of allowed operations may, therefore, vary considerably.

5.2 Query of a Spatial Database

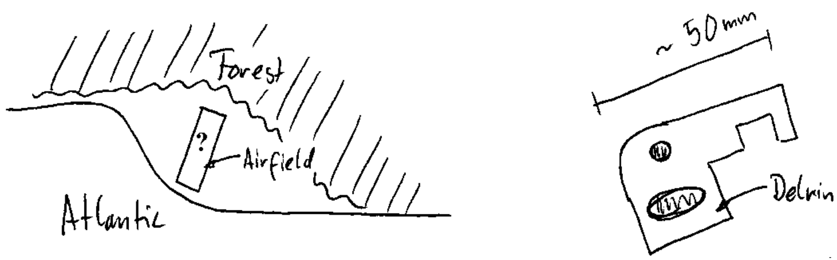
Compared with browsing, querying a database is considerably more specific right from the beginning, because a user must have an initial idea of what he or she is searching for, in order to be able to formulate a question. Unlike ordinary questions in our every day life, database queries are expressed in well defined query languages with a fixed syntax that is incompatible with our mental model of a question. When spatial components are involved, then there is a significant gap between a persons mental model and the final form of a question in a query language. To close this gap, we need a computer-understandable representation of a question that comes close to the way people think, a method, for example, that combines the simplicity of data browsing using gestures with the power of one of today’s fully developed database query languages.

We argue that sketching is an excellent form of expression for this purpose. Sketching is capable of covering the whole spectrum from standard relational to spatial queries, because both verbal and sketched statements are supported, which means that the user has the choice of the modality in order to express his question. Verbal statements are preferably used to describe non-spatial attributes or properties of objects in a sketch, such as the specification of a specific quality, dimension, or a functionality. Sketching on the other hand is employed when locations, relations, structures, or spatial attributes such as shape, direction, or topological issues must be addressed. The introduction of a sketch for a query is appropriate whereever a question can be formulated graphically. This kind of query is most likely the case when the initial idea of the question in a users mental model is spatially oriented.

The central issue of querying with a sketch is the retrieval of similarly structured configurations of objects in a spatial database. From this perspective it is also appropriate to speak of a *Query-by-Sketched-Example*. There are three main motivations for a user to sketch a query, what leads to the following query categories, which differ primarily with the expected number of retrieved matches of the database query:

Question:	Number of expected results:	
◆ *Where is it?*	(1)	match
◆ *Does it exist?*	(0 - n)	match(s)
◆ *Where else is it?*	(1- n)	match(s)

The focus of a *Where is it?*-query lies on the retrieval of a particular object configuration, although it is possible that during the search more than one result will be encountered. The knowledge of the initial configuration can vary from being fragmented to complete, but its exact location is unknown. The location can be associated to the real world or to the location in the database itself; therefore, it is possible to retrieve objects or groups of objects in different spaces, including the geographic space. As Figure 6 illustrates, it is possible to search directly for an object configuration (Figure 6a) or indirectly (Figure 6b) over a known part of a whole object structure.



(a) (b)

Figure 6	Two *Where is it?*-queries: (a) This sketch was made in order to retrieve the location of a specific airfield with a particular spatial configuration; (b) a sketch of a "lost" work piece in a mechanical parts database that was drawn to relocate the original plans associated to this part.

A *Does it exist?*-query is in general less concrete then a *Where is it?*-query, because the existence of a result is uncertain and the exact details are unknown; therefore, any number of answers between *0* and *n* can be expected. It is possible to search for single objects with certain spatial characteristics, such as a particular shape, but it is also feasible to query for specific object compositions. The search of an appropriate location for the construction of a hazardous waste site, for instance, could be guided by a sketched query. Such a query could take all regulations and environmental issues into account, including non-spatial as well as spatial aspects. Due to the generality of a sketched query, objects can be drawn in pictorial representations or as a sketched symbol, while the set of spatial relations among objects is implicitly given by the arrangement of these objects on the drawing surface. Because sketching—conversely to constructing—is not an exact science, objects and their relations can be partially incorrect. This is a particular important characteristic of sketched scenes and hence must be considered carefully when interpreting spatial scenarios.

A sketch-based *Does it exist?*-query can be seen as a spatial extension of a common web-browser query. Both query types allow a search for combinations of somehow related or connected objects, but the capability of being able to specify spatial relations and constraints that go beyond simple logical operations, such as *and, or* and *not*, gives sketching a significant advantage over a web-browser query.

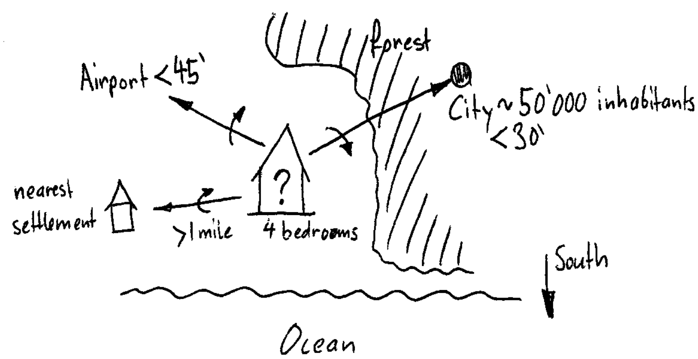


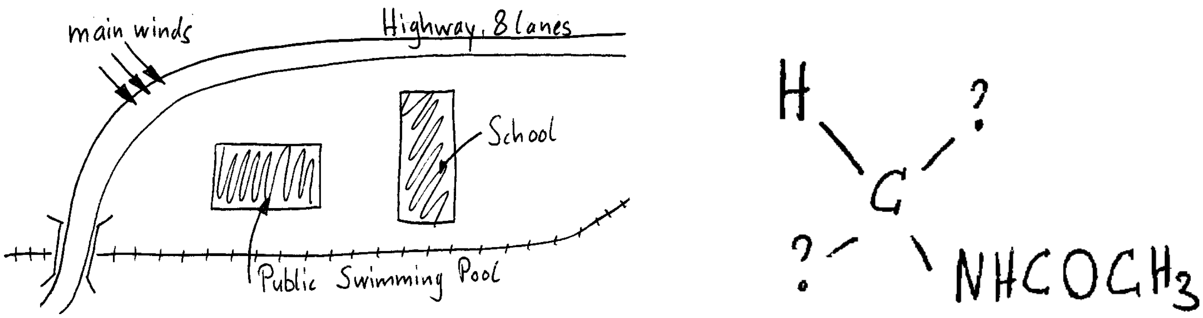
Figure 7	This example of a *Does it exist?*-query shows a sketch from a real estate agent, describing a customers specifications of a imaginary property.

The example on Figure 7 depicts a sketch of a real estate agent. The sketch was drawn to retrieve all properties that fit the needs of his customer. Although this sketch is rather simple, it contains a lot of information. The sketched objects are introduced with different symbolic representations. Drawing gestures are applied to specify distances, which are used in a direct (*<1 mile = distance*) and indirect (*<45' = time → distance*) manner. Alternative distances could have been declared by spoken language. Another drawn gesture—an arrow pointing south—indicates the orientation of the sketch. Geometrical constraints are declared implicitly with the chosen positioning of the property in question, the forest, the ocean, and the orientation with the south direction. However, this orientation has no importance for secondary objects, such as airport, town, or nearest settlement, because these objects are only used as qualitative descriptors. Finally the stipulation of attributes, such as *~50'000 inhabitants* or *4 bedrooms*, is done by writing them close to the appropriate objects—again, this could have been done also by spoken language.

The *Where else is it?* approach is similar to the *Does it exist?*-query. The main difference lies in the starting point, which is here an actual situation and not a hypothetical scenario. Hence, the primary focus of this query is to find configurations similar to an existent scenario. When the sketched query was reasonably drawn and when the sketched situation is part of the queried data set as well, then this initial situation should be included in the set of retrieved answers, too.

There are various possible applications for a *Where else is it?*-query. A typical field of application is the search for similar spatial structures in order to get insight into the relationship between the initial structure and its local environment. Another common task is the exploration of the database for object structures that share certain specific particularities of the initial structure. This search could also be entitled *The search for similar local* *spaces within the global space*. There are two fundamentally different types of local spaces: One possible type of local space is in itself complete, it neither overlaps with other local spaces nor can it contain other objects than the given ones. This absolute form of space is found mainly in elementary object configurations with clear boundaries and refers to closed queries. The second class of local spaces may overlap with other local spaces and there is no guaranty about the final composition of the retrieved scene, but that the retrieved local space is considered similar to the initial configuration as long as the spatial configuration of the objects and their properties match to a certain degree. Hence, additional objects that were not specified in the initial query can be present in the retrieved local scene as well.

This differentiation of closed and open queries applies also to *Where is it?-quer*ies. *Does it exist?-quer*ies are in general less concerned about closed spaces, because it is less likely that a user searches for elementary structures without knowing about their existence.



(a) (b)

Figure 8 Two *Where else is it?*-queries: (a) a problematic situation with a swimming pool, a school being squeezed between a highway, and a railway line (the intention of this sketch is to retrieve the locations of similar spatial configurations to get insight how such a situation could be mastered); and (b) a researcher has sketched an elementary substructure of chitin in order to query the database for other molecules containing this acetamid compound ($NHCOCH_3$).

The examples in Figure 8 show two *Where else is it?*-queries that demonstrate the difference between a closed and an open query. While results of query (8a) can include situations such that there is an other building between the school and the pool, it would be unacceptable for query (8b) if there was oxygen between the nitrogen and carbon element.

Querying is a principal user operation, particularly in systems with the purpose of managing and manipulating large amounts of data such as GISs—yet in every database application, querying is the most significant user operation. Hence, every simplification in the querying process is a great step towards an overall improvement of user-computer interaction, because like in many other modern systems, it is often the user who fails to accomplish an operation and not the machine. The introduction of a sketch-based user interface affects the process of querying a database in two ways: First, sketching facilitates the query procedure, because the user is allowed to use an additional modality that is easy and intuitive to apply and which is appropriate to translate the user's mental model into a commonly understandable form. Second, a host of queries are only possible if the spatial component is taken into account, what is extremely difficult if not impossible when using non-visual query languages, such as SQL.

The spectrum of possible queries, based on the three query types presented above and the potential field of applications, is hard to predict definitively, but it can be expected that applications far beyond GIS will benefit from sketching, once the sketching technology is mature and once the user community gets used to it. Possible applications can be anticipated for various scientific fields, such as geology, ecology, biology, and engineering in general, but also applications in the public interest area, such as law enforcement, national security, or defense, will benefit from this new style of querying a database.

5.3 Update of a Spatial Database

Updating a database is a substantial issue of the management part of taking care of a database. Data changes quickly and whether newly collected data must be added or already stored data must be altered, each database must be updated at a application specific interval to be up to date and to serve the purpose. There are basically to approaches to update a database: Automated procedures have been developed for many applications, examples are the automated object extraction from photogrammetric images, or the automatic generation of weather maps based on sample data. In other application fields an automated procedure is impossible since there are no suitable sensors available that can detect relevant changes or since available sensors are incapable to conceive changes within the required level of detail or with the specified accuracy. In these cases updates have to be carried out manually and often they have to rely on manual field notes. In such situations sketching may considerably simplify the update process, because the user can focus more on the his or her tasks and less on the translation of his or her observations into a computer understandable format.

Updates must be integrated into the context of the existing database. For objects in space this integration is done preferably in a graphical environment, such as a map, where spatial relations and features can easily be established by sketching them right onto the visual representation. The crucial part of such an operation—beside the interpretation of the sketch—is that the added or altered data must be consistent within the database before it can be definitively stored.

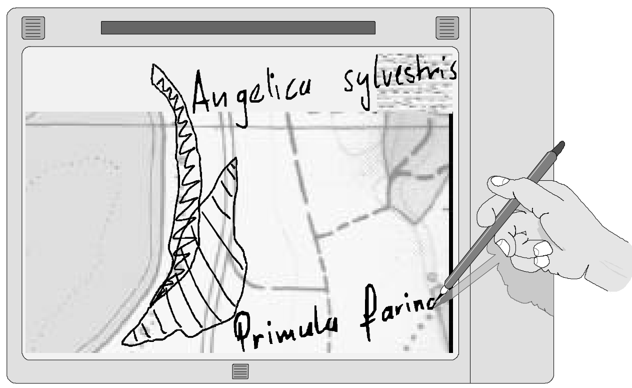


Figure 9 A section of a database in a map-like representation format. A ecologist is making a vegetation survey and sketches areas of spreading of species directly on the map. This information is analyzed and verified, and then written into a database record.

An update operation can involve other modalities beside sketching, such as pointing in order to select objects, writing, or talking to specify characteristics of objects or gestures in order to delete or to move selected objects. A database update involves a highly dynamic database where the characteristics of every single object can be addressed and object relations are mainly managed automatically. Applications that can benefit from such a setup range from simple data acquisition tasks used, for instance, in ecology (Figure 9) to more sophisticated assignments that involve multiple input sources or where it is necessary to describe 3-dimensional objects in application fields such as geology or petrography.

6. Conclusions and Future Work

In this paper we investigated the basic differences between today's graphical user interfaces and a sketch-based user interface. Beside human *modalities* and elementary computer interactions, we have explored more complex *user operations* and some fields of application where a sketch-based user interface can be successfully employed. We found that the set of *elementary actions* that a user wants to perform within an application remains unchanged. The five basic actions are: *Perceive*, *select*, *compose*, *change view*, and *operate*. On the other hand we have demonstrated that the way how these actions are employed in a sketching user interface differs significantly from a traditional user environment. While *pointing* has already become an important modality over the last 15 years, *drawing*, *talking*, and *hand-writing* will be the dominant modalities in a sketch-based user interface, limiting the use of a keyboard to exclusively text-oriented applications. Drawing can be divided into *drawing gestures*, which can be applied all over the interface, and *interpreted sketches* that are analyzed by the system and used for specific user tasks to express spatial, hierarchical, or conceptual problems, ideas, or intentions. Spoken verbal communication is primarily used to furnish the system with supplementary information while writing and drawing, and to help operating the system. Talking has, therefore, a supporting and—when used in a redundant way—stabilizing character (Egenhofer 1996a), although it can also be used as only modality for tasks, such as text generation.

Direct and intuitive interactions are key issues in a sketch-based user interface. The usage of drawing and talking assure easy-to-use systems that are still powerful and effective to operate. Considerable advantages can be expected for three database-related user operations in a GIS, which comprise *browsing*, *querying*, and *updating*. While the benefits for browsing and updating operations consist primarily of a simplification, querying experiences a boost of expressiveness. Hence, spatial queries that are difficult to formulate in a standard spatial query language can be reduced to a simple sketch, which includes all the necessary information so that the query can be processed—thus leaving the translation and interpretation part of the user's mental model to the system. This leads to the conclusion that sketching and talking are welcome and appropriate modalities in a GIS, particularly when spatial queries have to be formulated.

Subsequent research has to be conducted in two initially separate directions, one focusing on sketches and the other on spoken language. The goal of the initial research is to get insight about spatially motivated sketching and talking habits. Reoccurring and common expression patterns, gestures, and popular symbolisms have to be investigated and users with different origins, language or educational backgrounds must be taken into account. The chronology and natural sequence of spatial statements represent other potentially important issues to consider. Later phases will focus on the detection of relevant objects, their attributes in spatial expressions as well as on relations among objects. The gained knowledge about a sketch can then be used to specify a formal model of a sketch that can be further translated into a spatial query language.

7. Acknowledgment

This work was partially supported by Rome Laboratory under grant no. F 30 602-95-I-0042. The paper was born and formed throughout many valuable and creative discussions with my advisor Max Egenhofer. Thanks also to Yves Dennebouy, how helped revising the paper and provided precious advice for writing this paper.

8. References

Borning, A., H. (1979). ThingLab -- A Constraint-Oriented Simulation Laboratory. Stanford, Stanford University.

Borning, A. (1986). Defining Constraints Graphically. Human Factors in Computing Systems, CHI `86, Boston, MA.

Chok, S. S. and K. Marriott (1996). Automatic Construction of User Interfaces from Constraint Multiset Grammars. 11th International IEEE Symposium for Visual Language, Darmstadt, Germany.

Citrin, W. and M. D. Gross (1996). Distributed Architectures for Pen-Based Input and Diagram Recognition. Proceedings of the ACM, AVI '96 Gubbio Italy, Gubbio Italy.

Del Bimbo, A. and P. Pala (1995). Visual Image Retrieval by Elastic Matching of User's Sketches. 8th International Conference on Image Analysis and Processing, IAPR 95, September 13-15, 1995, Sanremo, Italy.

Egenhofer, M. (1992). "Why not SQL!" International Journal of Geographical Information Systems 6(2): 71-85.

Egenhofer, M. (1994). "Spatial SQL: A Query and Presentation Language." IEEE Transactions on Knowledge and Data Engineering 6(1): 86-95.

Egenhofer, M. (1996a). Multi-Modal Spatial Querying. Seventh International Symposium on Spatial Data Handling, Delft, The Netherlands.

Egenhofer, M. (1996b). Spatial-Query-by-Sketch. VL '96: IEEE Symposium on Visual Languages, Boulder, CO, IEEE Computer Society.

Egenhofer, M. and J. Herring (1993). Querying a Geographical Information System. Human Factors in Geographical Information Systems. D. M.-S. a. H. Hearnshaw. London, Belhaven Press: 124-136.

Gross, M. (1994). Strech-A-Sketch: A Dynamic Diagrammer. IEEE Symposium on Visual Languages, St. Louis, MO, IEEE Computer Society Press.

Gross, M. D. (1994). Recognizing and Interpreting Diagrams in Design. Advanced Visual Interfaces '94, ACM Press.

Gross, M. D. (1996). The Electronic Cocktail Napkin - computer support for working with diagrams.

Kato, T., T. Kurita, et al. (1992). A Sketch Retrieval Method for Full Color Image Database—Query by Visual Example. 11th IAPA International Conference on Pattern Recognition, The Hague, The Netherlands, IEEE Computer Society Press.

Kuhn, W. (1992). Paradigms of GIS Use. Fifth International Symposium on Spatial Data Handling, Charleston, SC.

Larkin, J. and H. Simon (1987). "Why a Diagram is (Sometimes) Worth Ten Thousand Words." Cognitive Science 11: 65-99.

Mark, D. and M. Gould (1991). "Interaction with Geographic Information: A Commentary." Photogrammetric Engineering & Remote Sensing 57(11): 1427-1430.

Medyckyj-Scott, D. and H. Hearnshaw (1993). Human Factors in Geographical Information Systems. London, Belhaven Press.

Neal, J., K. Bettinger, et al. (1988). An Intelligent Multi-Media Human-Computer Dialogue System. Workshop on Space Operations Automation and Robotics (SOAR 88), Dayton, OH.

Piaget, J. and B. Inhelder (1967). The Child's Conception of Space. New York, NY, Norton.

Spence, R. (1974). "MINNIE, a new direction in circuit design." Electronics Weekly, 3 July 1974.

Spence, R., P. Cheung, et al. (1986). "MINNIE: the way ahead for analogue CAD." Silicon Design, March 1986.

Sutherland, I. (1963). SketchPad: A Man-Machine Graphical Communication System. AFIPS Spring Joint Computer Conference.