# Multilingual Information Processing

**Gr. Thurmair,**
**Munich**

## 1.0 Overview

The following is a description of a system which will support multilingual information processing by applying techniques of natural language. Application fields can be medical engineering, drug enforcement, office document management, and the like.

These applications contains several technical challenges to the natural language processing components which go beyond standard machine translation:

- we have to cope with textual databases in several languages. If we want to query them, some translation process will be involved.
- we have to cope with the fact that we have to query not just textual but also structured information. Some translation into SQL type queries must be foreseen.
- there may be other objects as well in the database, like videos, audio tapes, and the like.

The search request should be stated in the users' native language, in natural language form, and the result should be available again in the users' native language.

Because of the multilingual aspect of such a system, special care needs to be taken to the question of multilinguality; it must be clarified which tools and multilingual techniques must be foreseen in such an application.

## 2.0 Workflow

The system aims at supporting the process of information gathering by accessing different and heterogeneous information sources. It follows the workflow of such an information processing task.

1. The first step is the **composition of the user request.** Users need to know which resources are available, and which information is relevant for a given request. This phase will be supported by offering the relevant information sources. The support consists of two steps:

- Search request building. Here, users will be able to select index terms, navigate through the domain model and other links between terms, compose information items and formulate an optimal search request.

- Search request analysis and decomposition. In this phase, the search request is broken into search items, enriched by additional search terms (on user request), and translated into the languages needed.

2. The second step consists in the **retrieval** proper. The search query will be decomposed into query elements; these query elements are searched in the respective databases. They may consist of structured elements and of textual elements. For structured elements, some SQL translation of the request must be foreseen. For textual elements, two ways of search will be performed:

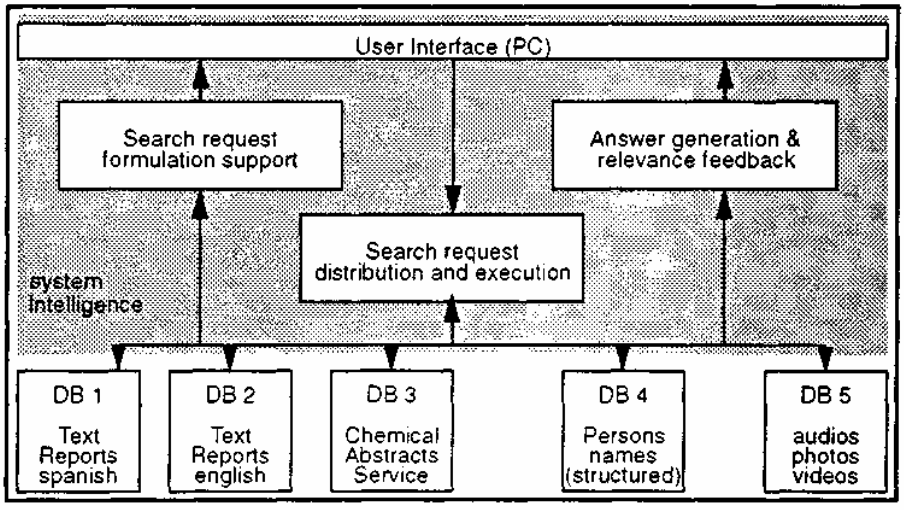
- fuzzy textual match, comparing the input with similar text descriptions;
- standard retrieval queries forwarded in natural language; no complicated retrieval language needs to be used.

In both cases, the search requests and text portion matches require a translation of the request into the language of the textbase to be searched. To do so, a terminological component has to be added to the system where the relevant terms can be translated into the target languages.

3. The third step consists in recomposing the retrieved elements of the query into meaningful statements **answering** the search request. This step implies retranslating the found elements into the language of the query, some answer formulation strategy based on the type of request, and a composition of the retrieved items into such an answer. The result should be a kind of report in the users' native language, containing all the information found.

Figure 1 shows the basic workflow concept.

**FIGURE 1. System Workflow**



## 3.0 System components

The system will run in a client server mode, offering a PC based user interface for search request formulation and answer generation. The user interface consists of the following elements:

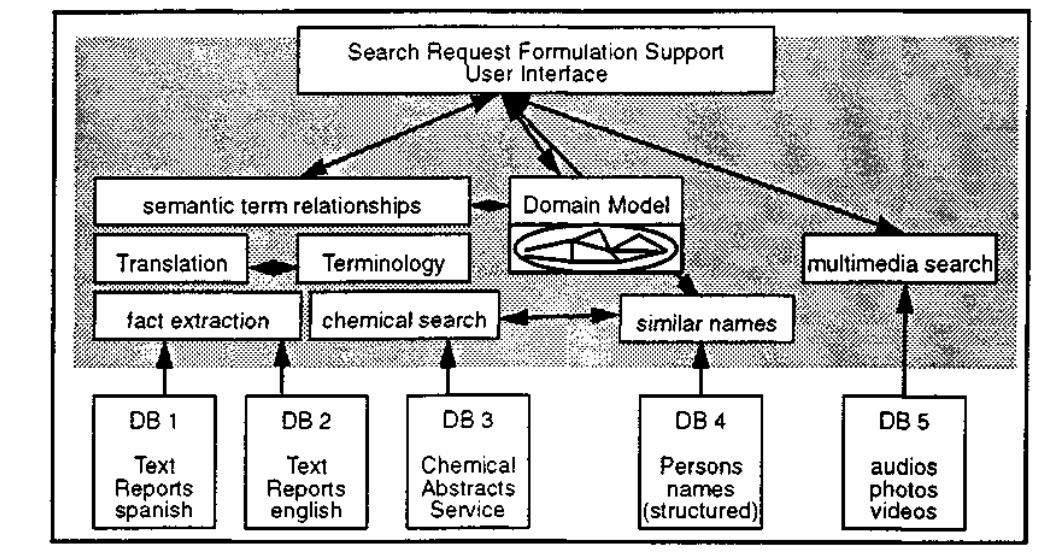
### 3.1 Support for query formulation

The basic retrieval means is a search form, combining structured and textual input possibilities. Users can specify their search request by filling in such a form. Aside from the fact that they can use their native language to do so, there
are the following possibilities in this scenario:

- inputting natural language can range from a simple noun phrase type of query (e.g. *recent letters to IBM about speech understanding products)* to inputting complete texts (e.g. medical files to be compared with similar records of a medical history).
- some formal parameters can also be given, e.g. in which database to search
- there will be several components to support users in their search. They will be the result of fact extraction techniques as well as linguistic processing of the text.

    - a component to recognise proper names in texts will identify these names, and present them to the user as search terms. As names may be misspelt or misunderstood, a component to identify similar names than the ones specified will be included. This component can be called by users as an additional help option.
    - a component to specify syntactically or semantically similar terms will also be available. This component is based on a linguistic analysis of dependency relations in texts, and presents networks of similar terms to the users. Again, users can look up these similar terms and add them to the search request, in order to make it more precise.
      It is a matter of the design whether these terms should be offered in the language of the database where they originate, or should be offered in a translated form, in the users' native language.
    - a third help function consists in a model of the domain. The idea is to have some domain specific relations between terms that can be used in the formulation of the search request *(e.g. persons -> offices -> companies* or *persons -> voices /photos).* The underlying component is a complex semantic network which serves as a basis for navigation.
    - There may be additional functions which could support the search request formulation. We could imagine a special component for searching chemical substances in special chemical databases (like Chemical Abstracts Service).

**FIGURE 2. Search Request Formulation**




These help functions can be called by users to collect search request items and to input an improved search request to the system. Figure 2 shows some of the components just mentioned.

From the system architecture point of view, the following components are needed in order to support this request formulation:

- a natural language query analyser for simple noun phrase type queries; it must convert the search term into a meaningful linguistic structure
- a natural language analyser in order to convert complete input texts into a searchable structure. Such a component could be taken from the analysis component of a translation memory system; maybe a full translation system is required.
- a text analysis component, identifying facts, like dates, proper nouns, and others, in a given database
- a text analysis component, identifying semantically similar terms in a corpus
- a component to analyse and generate similarities between proper names
- a domain model representation, to offer meaningful relations between information items at search time

Some of these components, esp. the text analysis components, require considerable linguistic machinery and some text corpora to run the analysis on.

In addition to the components mentioned, some terminology databases are needed in order to offer multilingual access. They are needed for simple term replacement operations as well as for more complex translation tasks.

There may be other objects to be searched for, like photographs, video or audio information. These objects also should be offered in the search request formulation.

Some of the components mentioned must be administered (e.g. a text processing run must be started, updates must be possible). For these tasks, a special administration interface is needed. This interface, however, will be distinct from the users' interface, and run on the server machine.

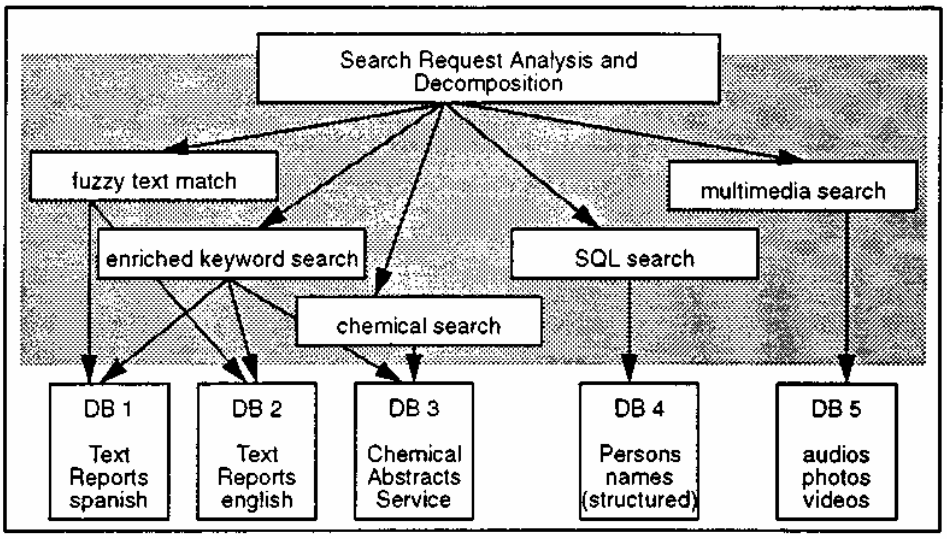
## 3.2 Search request decomposition and execution

A search request forwarded to the system will consist of a complex expression combining several basic information items. This request has to be processed. Several steps are needed to do so:

1. Identification of the information sources involved. Here it must be decided if a textual database should be searched, or if a request should be converted into an SQL type of query. Also the system explores its knowledge about which databases have to be accessed.
2. When preparing the DB access proper, the issue of translation becomes important: The search request, forwarded in the users' native language, must be launched in the DB's language. This implies some translation effort, ranging from simple term replacement to full machine translation.
3. In case of a full text being a search request, fuzzy matching techniques will be applied to identify the most similar texts. This approach may prove to work fine if there are many types of texts of a similar kind (like medical records). The matching will need a two-step search when large amounts of data are involved.
4. In case multimedia objects like audio or video information is searched for, the special retrieval possibilities of the databases in question have to be used.

An overview of the different techniques is given in Figure 3.

**FIGURE 3. Search decomposition and execution**



The search proper will access the best database available (or several of them),
and perform a search according to the functionality and possibilities of the respective database. This may even imply access of a remote database over network.

The result of the search will be (sets of) hits for each information item forwarded in a search request. These items have to be retranslated" into the users' conceptual world and native language. This "raw output" has to be reviewed, combined and translated into the output which the system is supposed to deliver.

## 3.3 Answer formulation support

The answer to a search request can itself be presented in several matters:

1. As far as multimedia hits are concerned, the system should have a possibility to display them, using special windows, or maybe even special hardware which standard multimedia kits offer today.
2. As far as structured items are concerned, they could be either presented in some structured form (e.g. tables of some kind), or they could become part of a answering report which the system produces according to some profile of type of search request: Every search request type could trigger a special report type generator, which would be fed with information items found in the databases.
3. As far as textual information is concerned, we have again several possibilities:

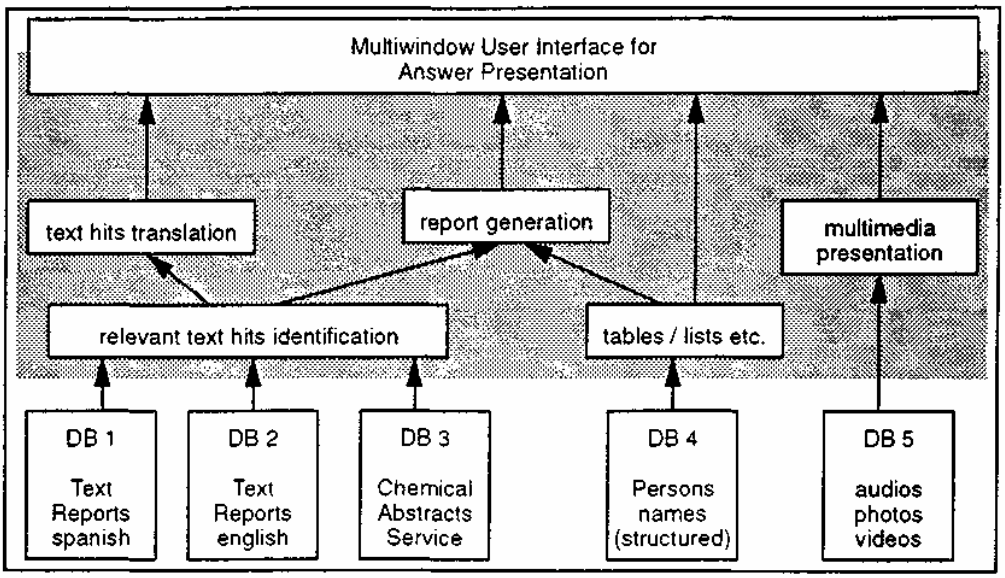
    - We can, after a fuzzy match, present the most similar text portions, the non-matching elements being marked (e.g. by colour) so that the user can easily identify identical and changed parts.
    - We also can simply present textual hits, as is done in conventional retrieval systems, the hit words being marked again.
    - If we can apply fact extraction techniques to the text such that only certain information items are relevant, these items could also be used as input for the reporting component just mentioned.

    In both cases, it must be taken into account that the language of the search request and the language of the text base may differ. Therefore we need a step of translation here again. This can be some memory based translation; it can also be full automatic translation (presenting the database language and the translated document parts in some split screen technique).

A special component is needed to compose the answers, to decide what should be presented in which order, and the like.

The main types of answer generation, and some of their interaction, are given in Figure 4.

**FIGURE 4. Answer generation and presentation**




Two types of operations should be possible on the basis of the answer:

- Updating the knowledge bases by marking relevant information items, and putting the information items into the appropriate slots of the knowledge base
- Using the search result as input for a further search request (relevance feedback); the current result may contain new or additional information, or may show that the original search request was not the best choice, so a new search request could make sense.

For both types of operations, the respective user interfaces have to be provided.

# 4.0 Aspects of Multilinguality

In the scenario just outlined, multilingual processing is one of the key issues for the success of such a system.

## 4.1 Multilinguality in Search Request Formulation

The language barrier has to be overcome on the following places here:

- As the search request can be stated in natural language (e.g. *"Eigenschaften des Ebola-Virus"),* the request must be translated into the language of the textual databases in question. Depending on the complexity of the request, simple term replacement operations may do, as by experience most search requests are NP-type structures. More complicated requests may need more sophisticated means of operation.
  Term replacement approaches face the difficulty of 1:n translation possibili-

ties (i.e. a given term translates into several target terms). By experience, wrong translations will be sorted out in the retrieval (there are simply no documents matching an odd translation).

- If whole texts are input in order to find the most similar texts, then these texts as a whole have to be translated. For this purpose, either an MT system could be used as we are in a rather homogeneous domain; this system would have to be tuned, however, with the target language structures in mind. The quality of the output depends on the quality of tuning. An alternative to a MT system could be a translation memory operating in a bi-lingual direction: from users' native to text base language for request formulation, and from text base to users' native language for answer generation.
- In the support of term search by linguistic extraction of similar or related terms, these terms also have to be presented in the users' native language. Moreover, an explicit term lookup facility should be offered. It is often a handicap in information retrieval, even for professional searchers, that the target language terminology is not at hand.

In all these cases, a component is needed which stores and maintains terminology, both for usage in other components and for lookup by the system users. This terminology component consists of a terminological database and an API for lookup (by other components and by the user interface).

We also may need a machine translation system if a full text is used as a search request. An alternative to this would be some translation memory tool, provided the text is highly standardised.

Of course the user interface needs other multilingual capabilities; e.g. the navigation in the domain model will have to be supported by translating the relevant nodes and links into the respective users' native language. This task, however, is similar to a "standard" localisation task.

## 4.2 Multilinguality in Search request execution

Once the search request is launched, it depends on the databases to be searched which operations must be performed:

- depending on the language of the targeted database, the respective information item must be translated into this language. Therefore, the translation (or term replacement) step can only be executed when the targeted text database is known: E.g. if a German request on *"Eigenschaften von Fluoriden"* is launched, it must be decided by the system that this request concerns the Chemical Abstracts Service database, and therefore has to be translated into English.
- The same argument holds for the text translation: A comparison of medical files with a Spanish database requires translation into Spanish, comparison with English requires an English translation, etc.

The tools needed here are the same as used in search request Formulation.

## 4.3 Multilinguality in Answer Generation

Answer generation requires some additional linguistic intelligence:

- If the search result is a (set of) textual hits, then these hits must be presented in the users' native language. In case of complicated texts, this task requires a machine translation system. Such a system has to do a on-line translation of all text hits. As MT systems do not translate perfectly, some split screen user interface could be given, allowing the users to check the translation and look up the original text if needed.
- In case of a fuzzy text match, the most similar text is presented, highlighting the non-identical text portions. If the fuzzy text match is in a foreign language then a complete translation memory based approach could be chosen. Such an approach requires multilingual alignment of such text portions, however.
- If an answer is to be given based on some report generation procedure, there are again several possibilities:

    - have several generation components which create natural language text from some formal representations. This would in principle be a monolingual task, to executed for several languages
    - generate one report in a pivot language, and translate this text into the respective users' native languages. If this is done by an MT system, we
      add the task of analysing the report to the task of running the different generation components. This technique is not superior to the one just mentioned except if some easier and more robust techniques can be used, based on translation memories.

## 4.4 Multilingual components needed

As a result, we need the full range of multilingual techniques which are known today:

- we need a term bank and terminology administration and lookup tool, in order to be able to offer multilingual term lookup and replacement
- we need a (bidirectional) translation memory to match and translate repetitive text and answer reports
- we need an MT system in order to translate complex search requests and text database hits.

All these techniques exist today. The challenge in the domain of information processing is to combine them with other linguistic techniques, in order to improve the overall system behaviour.

# 5.0 Literature


Andersen, P. M., Hayes, P. J., Huettner, A. K., Nirenburg, I. B., Schmandt, L. M., and Weinstein, S. P., 1992: Automatic extraction of facts from press releases to generate news stories. In: Proceedings of the Third Conference on Applied Natural Language Processing, pages 170--177. Association for Computational Linguistics, 1992.

Brown, P.F., et al, 1991: Aligning Sentences in Parallel Corpora, Proc. of the 29th Annual Meeting of the ACL, pp169-176, (1991)

M. Cavazza, and Zweigenbaum, P., 1992. Extracting Implicit Information from Free Text Technical Reports. Information Processing and Management, 28, 5.

Cowie, J., Wakao, T., Jin, W., Pustejovsky, J., and Waterman, S., 1993: The Diderot information extraction system. In: Proceedings of the First Conference of the Pacific Association for Computational Linguistics (PACLING 93), Vancouver, Canada, 1993.

Cahill, L.J., Gaizauskas, R., and Evans, R., 1992: POETIC: A Fully-Implemented NL System for Understanding Traffic Reports. In: Fully-Implemented Natural Language Understanding Systems: Proceedings of the Trento Workshop, March 30, 1992, pp. 86-99, IWBS Report No. 236, IBM Institute for Knowledge Based Systems, Heidelberg, 1992.

Damerau, F, 1993: Generating and Evaluating Domain-Oriented Multi-Word Terms form Texts, Inf. Processing & Management 29(4), 1993, 433-447

Evans, R., Gaizauskas, R. and Hartley, A.F., 1990: POETIC -- The Portable Extendable Traffic Information Collator in OECD Workshop on Knowledge-Based Expert Systems in Transportation, H. Jamsa (Ed.), Technical Research Centre of Finland, Espoo, 1990. Pp. 171-184.

Evans, R. and Hartley, A. F, 1990: The traffic information collator. Expert Systems: The International Journal of Knowledge Engineering, 7(4):209-214, 1990.

Fagan, J., 1989: The Effectiveness of a Non-Syntactic Approach to Automatic Phrase Indexing for Document Retrieval, JASIS 40(2), 1989, 115-132

Gaizauskas,R., Cahill, L. J., and Evans, R., 1993: Sussex University: Description of the Sussex System Used for MUC-5, In Proceedings of the Fifth Message Understanding Conference (MUC-5), Morgan Kaufmann, 1993.

Gaizauskas, R., Evans, R., Cahill, L. J., Richardson, J., and Walker, J., 1992: POETIC: A system for gathering and disseminating traffic information. In S.G. Ritchie and C.T. Hendrickson, editors, Conference Preprints of the International Conference on Artificial Intelligence, Applications in Transportation Engineering, pages 79-98, San Buenaventura, California, June 1992.

Gale W. A., Church K.W., 1991: A program for Aligning Sentences in Bilingual Corpora, Proc. of the 29th Annual Meeting of the ACL, pp 177-184, (1991)

Grefenstette, G., 1994: Explorations in Automatic Thesaurus Discovery, Kluwer Academic Publishers, Boston, Dordrecht, London, 1994

Jacobs, P.S., ed., 1992: Text-Based Intelligent Systems: Current Research and Practice, in Information Extraction and Retrieval, Lawrence Erlbaum, Hillsdale, NJ, 1992.

Jacobs, P. S. and Rau, L. F, 1990: Scisor: Extracting information from on-line news. Communications of the ACM, 33(11):88--97, 1990.

Kaji H., Kida Y., Morimoto Y., 1992: Learning Translation Templates from Bilingual Text, Proc. COLING(1992)

Kristensen, J.; Järvelin, K., 1990: The Effectiveness of a Searching Thesaurus in Free-Text Searching in a Full-Text Database, Int. Classification 17(2), 1990, 77-84

Lytinen, S. L, and Gershman, A., 1986: ATRANS: Automatic processing of money transfer messages. Proc. of the 5th National Conference on Artificial Intelligence (AAAI-86), pages 1089-1093, 1986.

McDonald, D.D., 1993. Internal and External Evidence in the Identification and Semantic Categorisation of Proper Names. Proceedings of SIGLEX workshop on Acquisition of Lexical Knowledge from Text pages 32-43, Ohio, U.S.A., June.

Raghavan, V; Wong, S., 1986: A Critical Analysis of Vector Space Model for Information Retrieval, JASIS 37(5), 1986, 279-287

Robertson, S.; VanRijsbergen, C.; Porter, M., 1981: Probabilistic Models of Indexing and Searching, in: Oddy, R.; Robertson, S.; VanRijsbergen, C.; Williams, P. (Hrg): Information Retrieval Research, Butterworth, London, 1981, 35-56

Ruge, G., 1992: Experiments on Linguistically Based Term Associations, Inf. Proc. & Management 28(3), 1992,317-332

Ruge, G.; Schwarz, C.; Warner, A., 1991: Effectiveness and Efficiency in Natural Language Processing for Large Amounts of Text, JASIS 42(6), 1991, 450-456

Sanderson, M., 1994: Word Sense Disambiguation and Information Retrieval, Proc. of SIGIR'94, Dublin, 1994, 142-151

Strzalkowski, T, 1994: Robust Text Processing in Automated Information Retrieval, Proc. of 4th Conf. on Applied NLP, Stuttgart 1994, 168-173

Sadler V., Vendelmans R., 1990: Pilot Implementation of a Bilingual Knowledge Bank, Proc. of COLING(1990)

Sato S., 1992: CTM : An Example-Based Translation Aid System, Proc. of COLING (1992)

Sato S., Nagao M., 1990: Toward Memory-based Translation, Proc. of COLING (1990)

Sumita E., Iida H., 1991: Experiments and Prospects of Example-based Machine Translation, Proc. of the 29th Annual Meeting of the ACL, (1991)

Wilks, Y, and Nirenburg, S., 1993. Large-scale knowledge base acquisition. In Proceedings of the Conference on very large knowledge bases. Tokyo,

Wilks, Y, Guthrie, L. and Farwell, D., 1993. The automatic acquisition of lexical entries for machine translation. Journal of Machine Translation.

Wilks, Y, 1991. Diderot: a text extraction system. In Proceedings of DARPA Speech and Language Workshop. San Mateo, Morgan Kaufmann, San Mateo, CA.