Cable Abstracting and INdexIng System (CANIS) Prototype

Ira Sider
Jeffrey Baker
Deborah Brady
Lynne Higbie
Tom Howard

Lockheed Martin Corporation

P.O. Box 8048
Philadelphia, PA 19101

sider,jbaker,brady_d,higbie, howard@mds.lmco.com

1. SUMMARY

The CANIS customer receives cables from sites world-wide, indexes the entities mentioned in these cables and stores that information for access by analysts at a later date. The CANIS customer indexes large quantities of information mostly manually, and wishes to reduce the human resources applied to this task.

Incoming cables are processed, information is extracted and stored in Corporate Databases. Cables with useful data are *abstracted* and *indexed.*

Abstracting captures information about the cable itself: its document number, source, date, etc.

Indexing captures information about the entities described in the cable: their names, dates of birth, locations, etc.

The result of the *abstracting* and *indexing* process is a set of index records about the entities that were described in a cable.

When a cable has been *abstracted* and *indexed*, its index record(s) are placed in a queue so that they will be stored. When file maintenance is performed (periodically overnight), the index records are stored in the Corporate Database.

The *abstracting* and *indexing* process is a time consuming and laborious task. Analysts must read every cable and extract the information that should be placed in the new index records or update existing records. Although the *abstracting* portion of the task has been automated, it is only a small part of the *abstracting* and *indexing* process. The majority of the effort is the *indexing* part of the process. At present, there is little or no automation support for the *indexing* part of the process.

The CANIS prototype is intended to assist the CANIS customer with the cable indexing task. CANIS automatically extracts entity information, builds and updates index records from cables, and presents it for review. CANIS' analysts can 1) approve the system generated index records, 2) add more information to the system generated index records, or 3) ignore the system generated index records and create their own. CANIS also extracts and stores relationship information (such as family relations, employment, and affiliations). This information is not currently identified or stored during the manual indexing task.

CANIS is compatible with both the input and output systems currently being used by the CANIS customer. CANIS runs on the customer-specified hardware and software platforms. However, the prototype CANIS system is a stand-alone system.

2. BACKGROUND

2.1. Concept Demonstration

CANIS Phase I Contract was completed February 1995. Lockheed Martin Management and Data Systems demonstrated the automatic extraction of the abstracted and indexed information from actual cables. Following the demonstration, M&DS developed requirements and design specifications for a prototype system that would meet the CANIS customers cable indexing and abstracting needs.

3. SYSTEM DESIGN

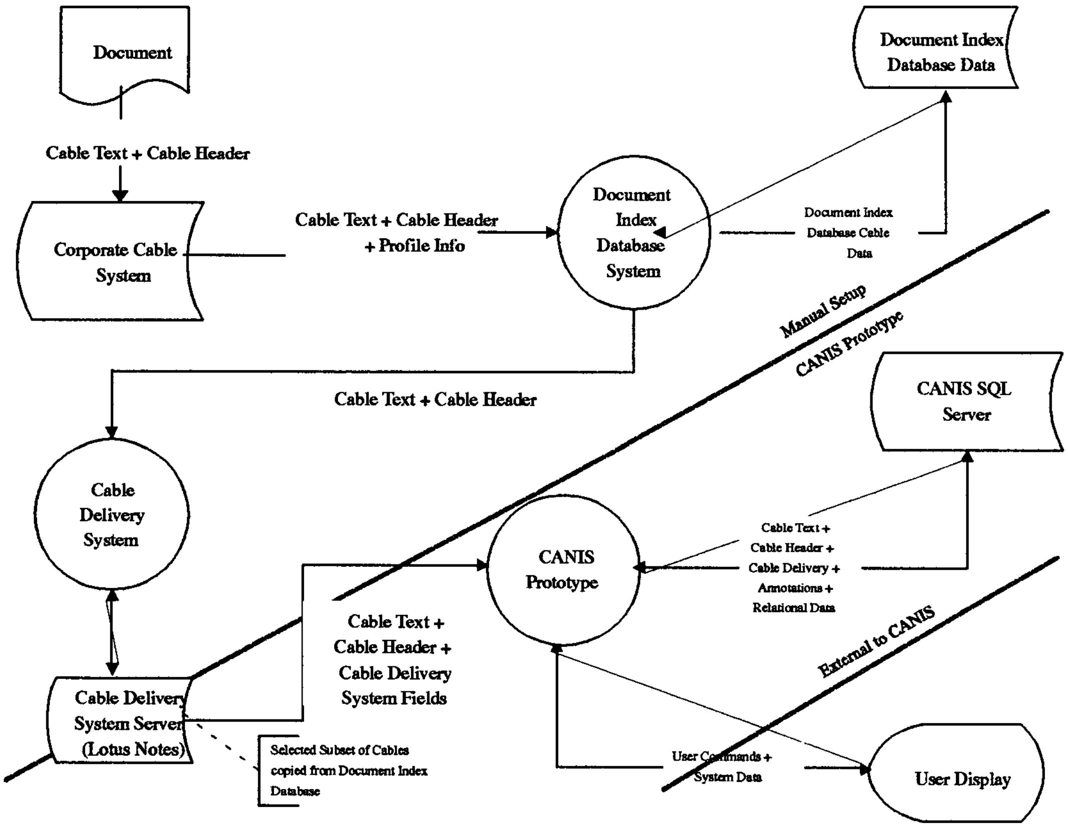
Document
Cable Text + Cable Header
Corporate Cable System
Cable Text + Cable Header + Profile Info
Document Index Database System
Document Index Database Data
Document Index Database Cable Data
Manual Setup
CANIS Prototype
Cable Text + Cable Header
CANIS SQL Server
Cable Delivery System
CANIS Prototype
Cable Text + Cable Header + Cable Delivery + Annotations + Relational Data
Cable Text + Cable Header + Cable Delivery System Fields
External to CANIS
Cable Delivery System Server (Lotus Notes)
Selected Subset of Cables copied from Document Index Database
User Commands + System Data
User Display

Figure 1.0 - CANIS External Interface Design

The CANIS prototype, as illustrated in Figure 1.0, will take as input, Cable Text, Cable Header, and Cable Delivery System Server Fields. CANIS performs all processing and stores the results internally. Users can visualize the processed data via the User Display.

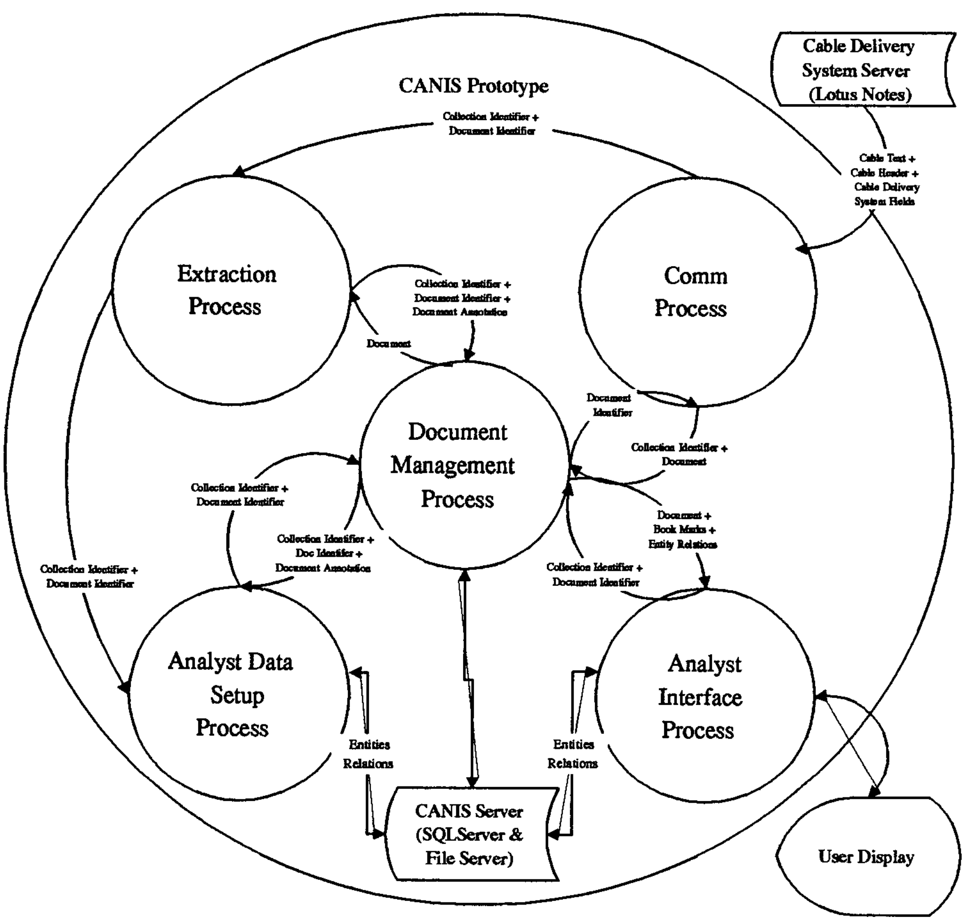
CANIS Prototype
Cable Delivery System Server (Lotus Notes)
Collection Identifier + Document Identifier
Cable Text + Cable Header + Cable Delivery System Fields
Extraction Process
Comm Process
Collection Identifier + Document Identifier + Document Annotation
Document
Document Management Process
Document Identifier
Collection Identifier + Document
Collection Identifier + Document Identifier
Collection Identifier + Doc Identifier + Document Annotation
Document + Book Marks + Entity Relations
Collection Identifier + Document Identifier
Collection Identifier + Document Identifier
Analyst Data Setup Process
Analyst Interface Process
Entities Relations
Entities Relations
CANIS Server (SQLServer & File Server)
User Display

Figure 2.0 - CANIS Top Level Process Overview

Figure 2.0 shows the top level design of the CANIS prototype. Cables are delivered to CANIS via the Cable Delivery System Server. This server acts as a communication driven pipe to the CANIS Prototype.

3.1. Comm Process CSCI

The Comm Process CSCI retrieves the Cable Data from the Cable Delivery System Server at a constant given rate via a software timer. The Comm Process CSCI creates a Document from the Cable Data and passes the Document to the Document Manager Process CSCI which stores this information in a Document Collection. The Comm Process CSCI sends the Collection Identifier and Document Identifier to the Extraction Process CSCI. The Comm Process CSCI also transfers the Cable Delivery System Header information to the SQL Database as it relates to the Document in the Collection.

3.2. Extraction Process CSCI

The Extraction Process CSCI processes each Collection Identifier and Document Identifier passed to it. The Extraction Process CSCI passes the Collection and Document Identifiers to the Document Manager CSCI which retrieves and returns the document text. The Extraction Process CSCI extracts biographical entities found within the document using Lockheed Martin's NLToolset. The Extraction Process CSCI passes the extracted entities to the Document Manager Process CSCI, which stores them as Annotations on the Document. The Extraction Process CSCI sends the Collection Identifier and Document Identifier to the Analyst Data Setup Process CSCI.

Upon initialization, the Extraction Process CSCI spawns the NLToolset Server Object, which ties all the NLToolset's data resources together into a single object, and loads into it the NLToolset System Specification File. This file contains a set of entries that identify the resources that should be loaded, the debug flags that should be set, the organization of the resources, and the sequence of operations that the NLToolset should perform. The primary resources that are loaded are:

- Lexicon
- Ontology
- Gazetteer
- Abbreviation
- Special Term List
- Set of Lexico-Semantic Rule Packages

The Extraction Process CSCI passes a Document through a series of NLToolset functions to perform the extraction. The steps are Tokenization, Segmentation, Reduction, Extraction, Reference Resolution, and Post Processing.

Tokenization creates a buffer of tokens from the Document's text. All words, punctuation, numbers, etc. in a Document are processed into tokens. The information captured for each token is: physical string, token symbol, token type, symbol id, part of speech, string case type, and character start and end positions.

Segmentation breaks the Document's token buffer into paragraphs and sentences based on multiple newlines, tabs, periods, etc.

Reduction performs multiple passes through the Document buffer looking for sequences of tokens that can be simplified into a single identifiable unit. These passes are used to identify specific pieces of information needed for Extraction.

Extraction and Reference Resolution are the NLToolset functions that glue all individual pieces together to create the entities. The information automatically extracted by CANIS from the Cables includes data of the following types:

- Person Names (All Types)
- Company/Organization Names
- Locations
- Dates
- Phone Numbers
- License Numbers
- Identification Numbers (example: Social Security)
- Gender
- Country of Birth
- Date of Birth
- Occupation
- Subject Line
- File Numbers
- Cable Numbers

The following associations will be extracted by the CANIS Prototype from the Cables:

- Family
- Employment
- Affiliations

Post Processing creates Annotations for all the data items and entities. These Annotations are then attached to the Document and stored in the Document Manager Process CSCI. Appendix A contains the Annotation Design Specification for these entities.

3.3. Analyst Data Setup Process CSCI

The Analyst Data Setup Process CSCI processes each Collection Identifier and Document Identifier pair passed to it. It then passes the Collection and Document Identifiers to the Document Manager Process CSCI which retrieves and returns the Document and its Annotations. The Annotations for this document are placed into relational records in the CANIS Server (SQL Server). Names, Organizations and Associations entities found within the extracted annotations are validated against existing entities. If the entity exists, then the new information is linked to the existing entities. If the entity does not exist, new relational records are created for that entity.

The Analyst Data Setup Process CSCI collects and builds relations for each of the major entities (Personnel, Organizations, and Associations) within the Annotations for the Document in the Collection. It validates and connects different types of locations, numbers and biographic information. For each entity, the process validates against existing index relations. If the entity exists, all information is processed as an update to the existing records. If the entity does not exist, the record is added to the relational database as a currently known Index record. Biographical entities are connected to the named entity through the ODBC SQLServer API. Biographical entity type connections are: gender, country of birth, date of birth, etc.

Additionally, the process operates on address entities and number entities and connects these to the named entities. It validates the address and number information against existing data relations. If the address or number exists, then all information is processed as a reference to the existing records. Otherwise, a new record containing the information is added to the relations and connected to the named entity. The types of addresses captured are: location, residence, etc. The types of numbers captured are: phone, license, etc.

The Analyst Data Setup Process CSCI links named entities together through relation links in the SQL database. The process will link the following entity information: Family (persons to family), Employment (persons to organizations), and Affiliation (persons to associations).

The Analyst Data Setup Process CSCI validates File Numbers against existing relations and connects them to named entities. The types of Filing and Document Reference data connected are: System Folder Objects and Document IDs.

Finally, the Analyst Data Setup Process CSCI adds the Document to an analyst working queue for processing by an analyst through the Analyst Interface Process CSCI.

The Analyst Data Setup Process bridges the gap between the information that was extracted from each Document and the information currently stored in the customer's database.

3.4. Analyst Interface Process CSCI

The Analyst Interface Process CSCI processes User Commands passed to it. These Commands allow an analyst to access and manipulate all the information stored in the CANIS prototype. When a Document is selected for display by an analyst, the Analyst Interface Process CSCI passes the Collection Identifier and Document Identifier for the Document to the Document Manager Process CSCI which retrieves and returns the Document and its relational records.

The Analyst Interface Process CSCI displays a summary list of the named entities associated with the selected Document. An analyst may select a given entity from the list and review the entity's detailed information, delete the entity from the list, or lookup a new entity found in the body of the Document. For each of the details available about a name, (ie. biographics, relationships, id numbers, locations, phone numbers etc.) the analyst reviews, modifies the information if necessary, and checks off the information. Some of the data, such as gender, citizenship, or relationship types, for example, have alternative choices available on a pull-down menu to minimize key strokes necessary to make changes.

The Analyst Interface Process CSCI allows an analyst to review and modify all information (Index records, addressees, subject line, and Filing locations) about a Document. It will display a Document's text body and allow the analyst to travel through the processing of the information about that Document. The following functions are available to an analyst: Document Details Review, CANIS Prototype Process Logs Review, Name Lookup and Processing, and System Filing.

Document Details Review displays the classification, addressees, and subject line associated with the current Document. The analyst may review and modify any of this information.

CANIS Prototype Process Logs Review displays the logs generated by each of the CANIS System Processes in read-only mode. The information captured by these logs includes: document identifiers for documents processed, error messages, system generated messages (ie. debug).

The Document Name Lookup and Processing allows the review and modification of named entities (Personnel, Company, and Associations) of the selected Document. The options available to an analyst here are, a) Name Lookup, b) Index, Review Data records for this entity, c) Create Links between Entity names and reviewed records; and d) Add and Modify Information associated with the entity (ie. gender, citizenship, locations, phone numbers, etc.)

Extraction errors found by the analyst during their processing of a document are appended to a log within the SQL database for review by an engineer for Extraction CSCI package adjustments.

3.5. Document Manager Process CSCI

The Document Manager Process CSCI is a set of library routines which provide a standard interface between the CANIS Prototype and the persistent storage of documents. The Document Manager conforms to the concepts and specifications of the TIPSTER Phase II Architecture Design Document (version 1.15). The Library routines of the Document Manager Process provides all CSCI's of the CANIS Prototype with a standard interface (API) for accessing documents, and communicating annotation information about those documents.

The Document Manager Process CSCI is implemented on top of a relational database with access to the database facilitated through ODBC library calls. The Document Manager Process CSCI uses Microsoft's ODBC library, Microsoft Access and Microsoft SQLServer. Other applications using Lockheed Martin's Document Manager are being built on top of Sybase and Oracle.

4. NLTOOLSET

The NLToolset is a framework of tools, techniques and resources for building text processing applications. The NLToolset is portable, extensible, robust, generic and language independent. The NLToolset combines artificial intelligence (AI) methods, especially NL processing, knowledge-based systems and information retrieval techniques, with simpler methods, such as finite state machines, lexical analysis and word-based text search to provide broad functionality without sacrificing robustness and speed.

The NLToolset currently runs on SUN Microsystem's UNIX-based platforms and PCs (using Microsoft Windows NT). The NLToolset is coded in C++ and uses the COOL Object Library. The CANIS application is PC based and using the Microsoft Visual C++ compiler and Visual Basic on the PC.

5. TESTING AND EVALUATION

We are currently in the testing phase and developing the evaluation criteria in conjunction with the government These phases are scheduled to complete, July 1996.

Our Test Plan involves subsystem testing of each of the Comms Process CSCI, Extraction Process CSCI, Analyst Data Setup Process CSCI, and the Analyst Interface Process CSCI. We are also performing System Level Integration Testing to validate the data passing through each process within the CANIS application.

Evaluation will be performed by analysts at their site using real data. We are currently working with the customer to determine evaluation criteria.

6. CONCLUSIONS

The CANIS prototype will show the customer a new way of doing business. Analysts will see their tasks change from manually reading and creating index records to verifying and updating automatically generated index records. Their daily process will involve more analysis than data entry and they will be able to process a larger number of documents in a single day.

7. REFERENCES

1. CANIS System Requirements Specification;
   May 30, 1995.
2. CANIS System Design Specification;
   November 30, 1995.
3. TIPSTER Phase II Document Manager Specification (v1.15)
   September, 1995.

APPENDIX A.
ANNOTATION DESIGN SPECIFICATION

```
annotation type r-fullname
annotation type r-descriptor
annotation type r-surname
annotation type r-birth-date
annotation type r-title
annotation type r-occupation-field
annotation type r-occupation-text
annotation type r-file-number
annotation type r-phone-number
annotation type r-org-name
annotation type r-country
annotation type r-address
annotation type r-city
annotation type r-id-number
annotation type r-gender
annotation type r-state
annotation type r-mailcode
annotation type r-subject-line
annotation type r-date
annotation type r-text
annotation type r-phone-text


annotation type c-tperson
    {r-fullname: r-fullname
     r-variation: sequence of r-fullname
     r-descriptor: r-descriptor
     r-surname: r-surname
     r-gender: r-gender
     r-birth-date: r-birth-date
     r-taddress: sequence of c-taddress
     r-tphone: sequence of c-tphone
     r-tidnum: sequence of c-tidnum
     r-title: sequence of r-title
     r-occupation-field: r-occupation-field
     r-occupation-text: sequence of r-occupation-text
     r-text: sequence of r-text
     r-person-type {e.g., MAIDEN ...}
     r-associated {Y}
     }
```

```
annotation type c-taddress
    {r-address: r-address
     r-city: r-city
     r-state: r-state
     r-country: r-country
     r-mailcode: r-mailcode
     r-address-type: {BIRTH, LOC, ADDRESS}
     r-associated {Y}
     }

annotation type c-tphone
    {r-phone-number: sequence of r-phone-number
     r-phone-type: {PHONE, FAX, TELEX}
     r-phone-text: sequence of r-phone-text
     r-associated: {Y}
     }

annotation type c-tidnum
    {r-id-number: sequence of r-id-number
     r-id-type: {SSN, LICENSE}
     r-associated: {Y}
     }

annotation type c-torganization
    {r-org-name: r-org-name
     r-variation: sequence of r-org-name
     r-org-type: {COMPANY, GOVERNMENT,
         OTHER}
     r-descriptor: r-descriptor
     r-taddress: sequence of c-taddress
     r-tphone: sequence of c-tphone
     r-associated {Y}
     }

annotation type:  c-tfamily
    {r-fullname: r-fullname
     r-surname: r-surname
     r-taddress: sequence of c-taddress
     r-tphone: sequence of c-tphone
     r-associated {Y}
     }


annotation type c-parent-assoc
    {r-child: c-tperson
     r-parent: c-tperson
     r-descriptor: r-descriptor
     }

annotation type c-father-assoc
    {r-child: c-tperson
     r-father: c-tperson
     r-descriptor: r-descriptor
     }
```

```
annotation type c-mother-assoc
      {r-child: c-tperson
       r-mother: c-tperson
       r-descriptor: r-descriptor
       }

annotation type c-brother-assoc
      {r-sibling: c-tperson
       r-brother: c-tperson
       r-descriptor: r-descriptor
       }

annotation type c-sister-assoc
      {r-sibling: c-tperson
       r-sister: c-tperson
       r-descriptor: r-descriptor
       }

annotation type c-sibling-assoc
      {r-sibling-a: c-tperson
       r-sibling-b: c-tperson
       r-descriptor: r-descriptor
       }

annotation type c-married-assoc
      {r-spouse-a: c-tperson
       r-spouse-b: c-tperson
       r-descriptor: r-descriptor
       }

annotation type c-family-other-assoc)
      {r-person-a: c-tperson
       r-person-b: c-tperson
       r-descriptor: r-descriptor
       }

annotation type c-family-member-assoc
      {r-family: c-tfamily
       r-family-member: c-tperson
       r-descriptor: r-descriptor
       }

annotation type c-maiden-persona-assoc
      {r-person-a: c-tperson
       r-person-b: c-tperson
       r-descriptor: r-descriptor
       }

annotation type c-other-persona-assoc
      {r-person-a: c-tperson
       r-person-b: c-tperson
       r-descriptor: r-descriptor
       }

annotation type c-employment-assoc
      {r-person: c-tperson
       r-organization: c-torganization
       r-descriptor: r-descriptor
       }

annotation type c-affiliation-assoc
      {r-person: c-tperson
       r-organization: c-torganization
       r-affiliated-with: c-torganization
       r-descriptor: r-descriptor
       }


annotation type c-tdocument
      {r-subject-line: r-subject-line
       r-reference-line: r-reference-line
       r-categories-type: sequence of {DRUGS,
            POLITICS, TERRORIST, OTHER]}
       r-associations: sequence of c-affiliation-assoc or
            c-employment-assoc or c-contact-assoc or
            c-persona-assoc or c-family-member-assoc
            or c-family-other-assoc or c-married-assoc
            or c-sibling-assoc or c-sister-assoc or
            c-brother-assoc or c-mother-assoc or
            c-father-assoc or c-parent-assoc
       r-unassocpersons: sequence of c-tperson
       r-unassocorgs: sequence of c-torganization
       r-unassocaddr: sequence of c-taddress
       r-unassocphone: sequence of c-tphone
       r-unassocidnum: sequence of c-tidnum
       r-unassocfamily: sequence of c-tfamily
```