

The background of the image is a collage of various items: a magnifying glass with an Apple logo on its handle, a green pen holder with several pens, a globe, and some papers. The text is overlaid on this background.

Worldwide

Developers

Conference



Apple Information Access Toolkit

Dan Rose

**Manager, ATG Information
Access Research Program**

What AIAT Is

An Information Access Toolkit providing content-awareness to applications

- An actual product!
- A modular C++ library
- Designed to be platform-independent
- Designed to be language-independent
- Formerly known as “V-Twin”



What AIAT Isn't

- **Not an application**
- **Not something with an interface**
- **Not (just) a search engine!**



What Is Content-Awareness?

*Having the ability to support actions based on what a document is **about**, such as...*

- **Searching**
- **Filtering**
- **Browsing**
- **Summarizing**
- **Describing**
- **Organizing**
- **Filing**
- **And more...**



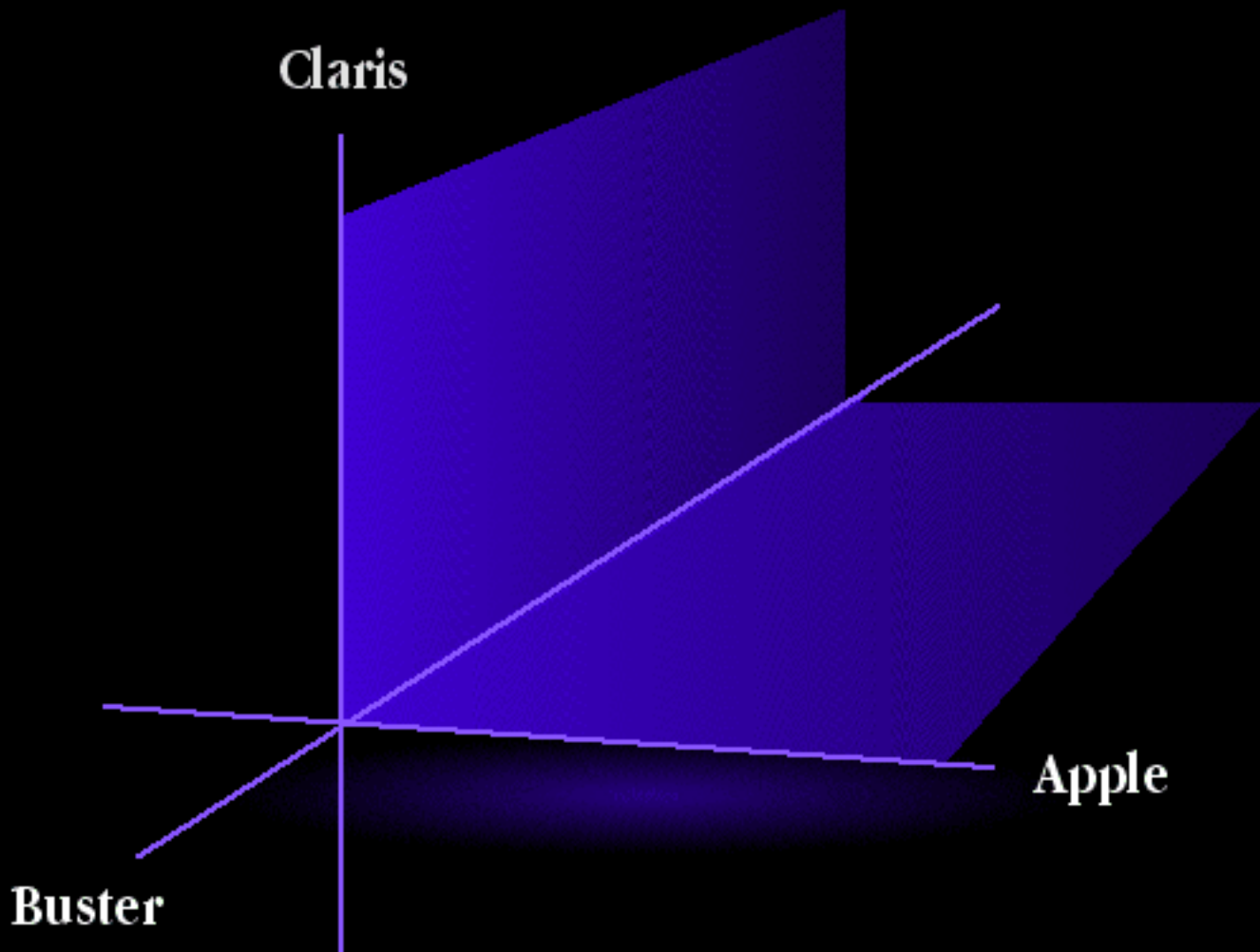
How It Works

We create a concise representation of what each document is about

- **Based on a statistical analysis of the distribution of words in the collection**
- **Representation may be thought of as a vector in a high-dimensional space**
- **All the operations boil down to figuring out how close items are in the space**



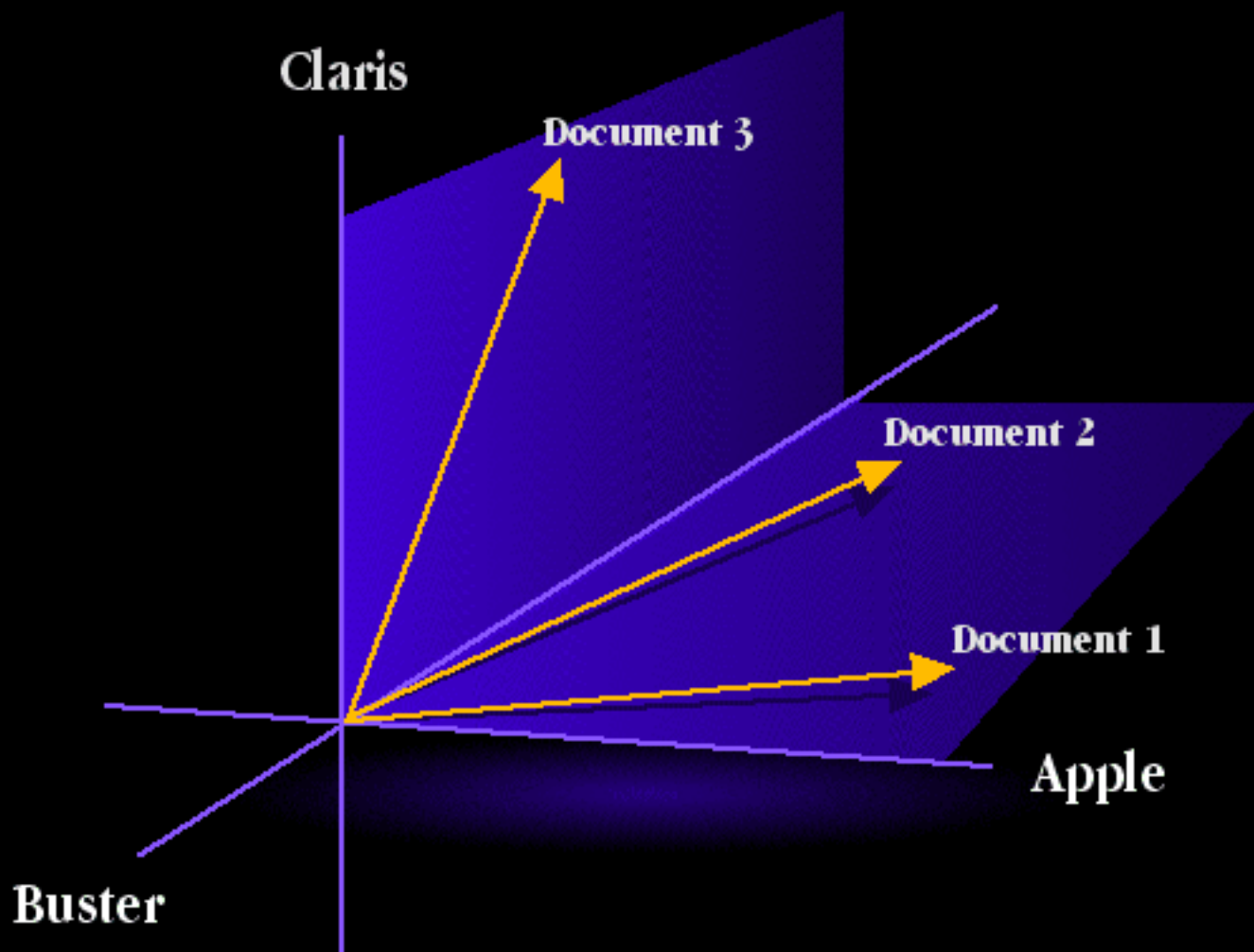
The Document Space



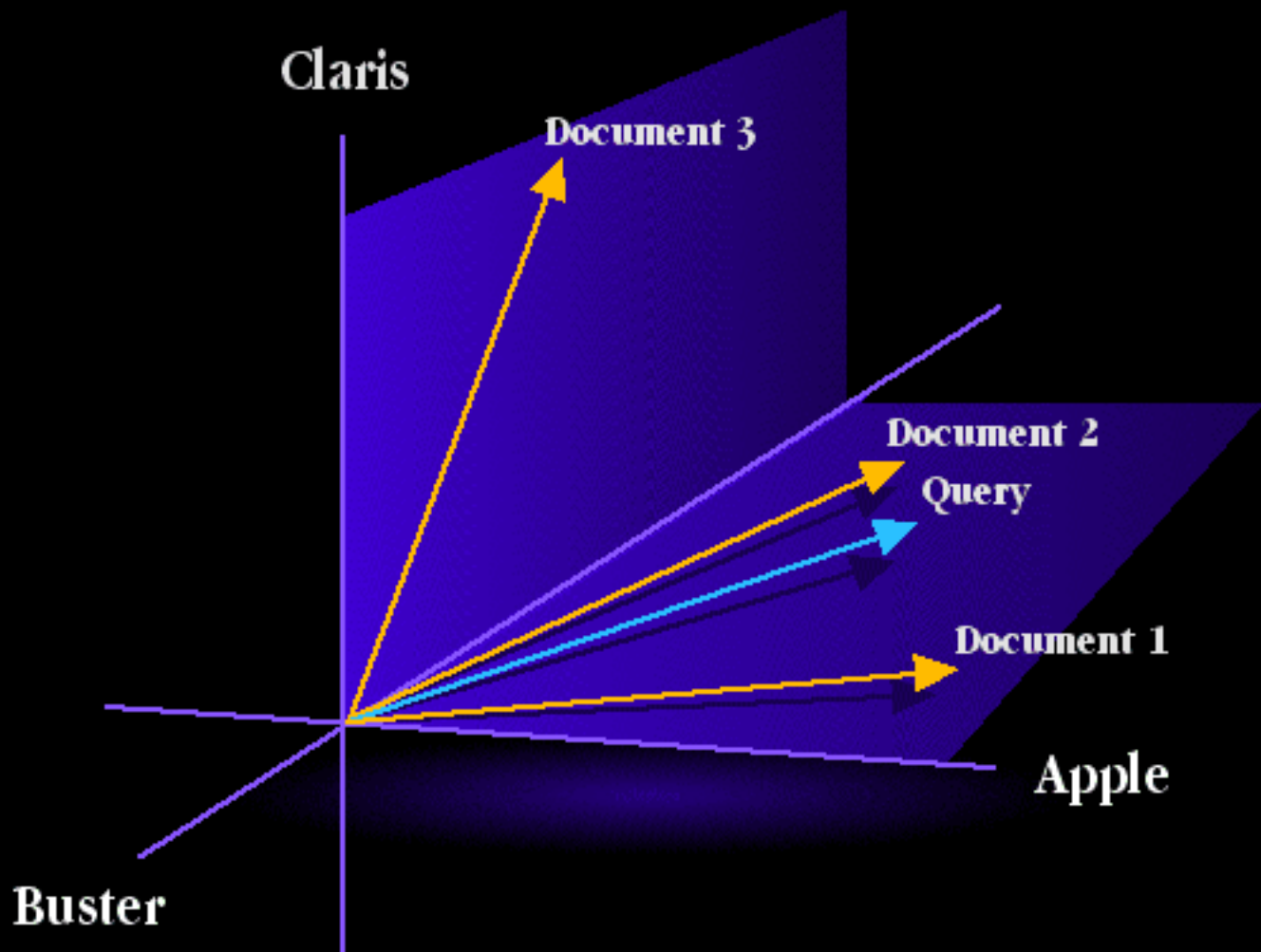
Buster

Apple

Document Vectors



Querying the Documents



Buster

Features

- **Low disk overhead**
 - 15–25% of source text, depending on features used
 - Index is updated in place
- **Low and scalable memory overhead**
- **Uses global context for more accurate relevance ranking**
- **Optimized for interactive use**



AIAT at Work

- **In Apple products...**
 - Cyberdog
- **In Apple tools and technology demonstrations...**
 - Apple e.g.
 - Apple HTML Local Search
- **In Apple prototypes...**
 - Vespa (summarization)
 - Kinshasa (clustering)



AIAT Architecture

Accessor

Index

Storable

Analysis

Storage

Corpus



Architecture: Storage

Persistent storage API for arbitrary objects

- Uses safe atomic transactions
- Supports multiple threads
- Typically used in conjunction with...



Storage

Architecture: Storable

Management of objects to be stored

- **IStorable**
 - Objects that know how to store themselves
- **IAOrderedStorable**
 - Objects that know how to store and sort themselves
- **IAOrderedStorableSet**
 - Collections of IAOrderedStorables with iterators and a variety of access methods



Storable

Architecture: Corpus

Definition of the set of documents

- **The corpus classes define what constitutes a document for your application. It may be a...**
 - File system file
 - Database record
 - E-mail message
 - Arbitrary range of bytes



Corpus

Architecture: Analysis

Transforms input text into index terms

- **Your application may use any or all of the following analysis modules**
 - Tokenization
 - Stemming
 - Stop word removal



Analysis

Example of Analysis Modules

Sending (the letters) to aiat@apple.com!



Example of Analysis Modules

Sending (the letters) to aiat@apple.com!



sending the letters to aiat apple com



Example of Analysis Modules

Sending (the letters) to aiat@apple.com!



sending the letters to aiat apple com



sending letters aiat apple com



Example of Analysis Modules

Sending (the letters) to aiat@apple.com!



sending the letters to aiat apple com



sending letters aiat apple com



send letter aiat apple com



Architecture: Index

Creates and maintains the concise representation of document contents

- Base classes do not assume words; data could be anything
- Supports efficient processing of both
 - Term-based operations (such as conventional search)
 - Document-based operations (such as similarity search)
- Index lives in storage, anywhere you want



Index

Architecture: Accessor

Used to access content representations stored in indexes

- Search
- Describe document
- Find related words



Accessor

Example: Creating an Index

instantiate the storage
initialize the storage

create the index—in the storage,
on a corpus, with an analysis
add documents to the index
flush index changes

commit the storage



Example: Creating an Index

```
myStore = MakeHFSStorage(vRef, dir, name);  
theStorage->Initialize();
```

```
theIndex = new InVecIndex(myStore,  
    new HFSCorpus(),  
    new SimpleAnalysis());  
theIndex->Initialize();
```

```
theIndex->AddDoc(anHFSDoc);
```

```
theIndex->Flush();
```

```
theStorage->Commit();
```



Example: Search with Accessor

instantiate the storage and index as before

open the storage

open the index

create an accessor on an array of indexes

initialize the accessor

use the accessor to search



Example: Search with Accessor

```
// (instantiate storage & index as before)

theStorage->Open();
theIndex->Open();

indexes[0] = &theIndex;
theAccessr = new InVecAccessor(indexes, 1);
theAccessr->Initialize();

byte *query = "new Mac OS technologies";

numHits = theAccessr->RankedSearch(
    query, strlen(query),
    NULL, 0,
    hitArray, maxHits, maxTermsPerDoc,
    &progrssFn, progrssFreq, progrssData);
```





Easy Rider:

*Content Search
on the Desktop*

Demo

Example: Prepare for Sorting

index all the existing documents

for each folder

 average the vectors for each document



Example: Prepare for Sorting

```
// (index documents as before)

TWVector *docVec;
TWVector *folderVec = new TWVector(0);

for (i = 0; i < numDocsInFolder; i++) {
    docVec = accessor->GetTWVector(doc[i], 0);
    folderVec = folderVec->Sum(docVec);
}

folderVec = folderVec->Normalize();
```



Example: Sorting New Files

for each new document
add document to index
compute similarity with each folder
move document to most similar folder



Example: Sorting New Files

```
// (add doc to index, make accessor)

docVec = theAccessor->GetTWVector(doc, 0);

bestFolder = kNoFolder;
bestScore = -1.0;

for (j = 0; j < numFolders; j++) {
    score = docVec->Similarity(folderVec[j]);
    if (score > bestScore) {
        bestScore = score;
        bestFolder = j;
    }
}

// now move doc to bestFolder
```





Conakry:

*Sorting Files
into Folders*

Demo

Developer Opportunities

- **Enhance your existing application with search or other capabilities**
- **Help manage Internet/Intranet information**
- **Develop custom AIAT modules, e.g., for other languages**
- **Help create a whole new class of content-aware applications**



AIAT Status and Availability

AIAT 1.0 currently available for licensing

- For more info...
 - Web: <http://devworld.apple.com/dev/aiat>
 - E-mail: aiat@apple.com





<http://devworld.apple.com/dev/aiat>

aiat@apple.com

Q&A

The background features a dark, textured surface with a glowing blue and purple sphere in the center. A white Apple logo is positioned at the top of the sphere. The text "Worldwide Developers Conference" is overlaid on the image. "Worldwide" and "Conference" are in a gold, serif font, while "Developers" is in a white, serif font enclosed in a white rectangular box. The overall aesthetic is futuristic and tech-oriented.

Worldwide

Developers

Conference