Worldmatch®
Comparing International Data

As more processes serving customers and suppliers are automated, uniform master data records become increasingly important. Even as recognition of this fact grows, challenges involved increase as well: data comparisons remain anything but trivial, and errors can become more expensive than ever before. International comparisons are especially critical: using the correct algorithm is paramount.

Comparing addresses with each other

It is frequently necessary to compare addresses against one another, such as when checking for duplicates, entering new customer data, searching for addresses in a call-center, or during international address comparisons.

The word "compare" sounds simple at first, but something that may be easy for a human can be very difficult for a computer. For an electronic brain, "equal" means 100 percent agreement. Even the smallest variation (such as any typographical error) will prevent a correct association. A computer will not recognize the similarity between "Matthias" and "Mathias", at least not without programmed intelligence. On the other hand, a human will instantly see that these two spellings are nearly identical. This kind of intelligence is a critical factor for electronic address comparisons.

Teaching similarity to computers

Clever programmers have been battling duplicate addresses since the 1960's. At first computing time was very expensive. For example, the flight of Apollo 11 in 1969 was calculated by a ground computer with a performance equivalent to the 286 CPU's of the 1980's (many desktop systems today have more than 50000 times as much computing power!) Thus for economic reasons, early comparison processes had to strictly limit their use of computing power. It was cheaper to accept losses caused by a few missed duplicates than to invest in a more powerful mainframe system. This ultimately led to the first quick and efficient "matchcode" algorithms.

Matchcode processing: quick, but not precise

Instead of comparing two addresses letter ▸

for letter, a matchcode algorithm compares only important isolated portions, such as the postal code, house number, the first and third letters of the last name, and the first letter of the first name. As long as all of these items match, the records will be treated as matching, even if other portions of the addresses are different.

This processing form requires minimal computer time, since the "matchcode" is easy to construct and can be stored in an index for every address as it is entered: the code does not need to be recalculated each time for every address. The weakness is very poor resolution: "Maier" and "Meier" will be associated correctly, but for "Meyer", the relevant third letter is different, and the matchcode method fails. Conversely, the names "Meyer" and "Mayfly" would incorrectly result in a positive test. For obvious reasons, matchcode algorithms are now obsolete.

Phonetic algorithms: Typos lead to disaster

A phonetic method unifies similar spellings by converting letters with similar sounds to equivalent codes. For instance, P and B could be represented as "1", and C, K, and G as "2". This allows "Becker" and "Beggar" to produce a "matching" phonetic result.

Just as with a matchcode process, phonetic codes can be stored in an index, accelerating the algorithm enormously. There are many different methods: the simplest work just on individual letters, whereas more sophisticated systems also consider groups of letters (such as "ph" or "th"). One simple method is the Russell-Soundex algorithm, usually referred to simply as „Soundex". It is very popular, but also produces many errors; for example, it will match "Mehl" and "Maier". For duplicate comparisons, such incorrect associations lead to so-called "overkill".

Modern phonetic systems will find complicated variations such as "Christopher" and "Kristofer" or "Klusoh" and "Clouseau", and are able to deal with foreign letters (such as accented or umlauted vowels). However, even the best phonetic system can only find phonetic errors. Typographical errors or abbreviations fall outside the net, and will not be discovered by such a system. This is why phonetic algorithms alone are simply insufficient (at least for corporate address data).

"Fuzzy" similarity algorithms: the better choice

Since computers have become more powerful and less expensive, it is no longer necessary to use indexed methods. Matchcode calculations have since been replaced by so-called "fuzzy" algorithms. A "fuzzy" comparison does not make a Yes/No decision, but instead determines a relative degree of similarity. By considering different elements in the address, a much higher resolution may be obtained. For instance, if both the company name and the individual contact name are very similar, two addresses may be identified as matching duplicates, even when ▶

Carsten Kraus, CEO of Omikron Data Quality GmbH

About Omikron Data Quality GmbH

It all began in 1992 with the creation of a new system for similarity evaluations in computer-based duplicate checks: the FACT® algorithm. It quickly developed that Omikron's new invention performed much better than existing matchcode and phonetic processes. In 1993, we created a software comparison program, which by 1996 grew to become a complete application set for managing the quality of address data. In 1997 and then again in 2003 we replaced the entire application set with newly updated generations for the complete portfolio, supporting the latest programming languages and development methods.

In 2007, Omikron Data Quality Server was released for Service Oriented Architecture (SOA) applications. This integrated solution allows companies to verify their Data Quality at every critical point within their entire IT infrastructure. In the same year our new Worldmatch® comparison algorithm set completely new standards for international data comparisons.

Today Omikron is a leading German provider of Data Quality solutions. Omikron integration modules are available for all popular business applications (including SAP® and Microsoft CRM®).

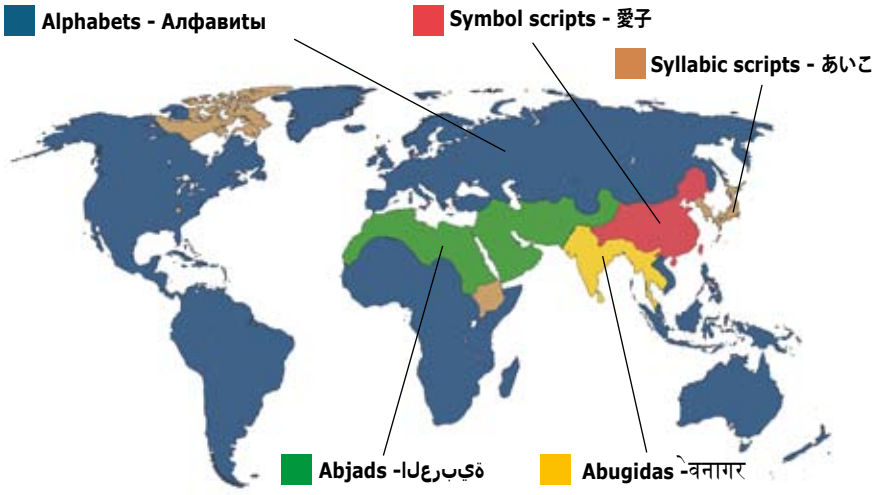

Illustration: Worldwide distribution of script systems

the street addresses are completely different (such as might happen when a company moves). If the street name and house number are virtually identical, then a stronger variation in the company name would be acceptable. This sort of weighted approach cannot be dealt with using matchcode processes.

Initial approaches such as "Levenstein" were directly entirely at typographical errors, but modern algorithms achieve much more. For example, Omikron's FACT® algorithm finds all of the examples listed above, and thus comes extremely close to a human ability in recognizing similarities.

Worldwide data comparisons: the immediate challenge

In 1993, Germany achieved 20% of its gross domestic product through export sales. Today this figure is over 30%, and the process of globalization is accelerating.

Merging economic systems have enormously increased demands on every company's corporate data: a firm with its headquarters Europe will use the Roman alphabet, but a branch office in Morocco will require Arabic. A company that wants to be a global player needs a database infrastructure and methods that are equally applicable worldwide, because even when logistics permits markets to combine physically, there is little hope that the world will settle any time soon on a single language and a unified character set for all communications.

Internationalization makes it more difficult to work with data, because a comparison must be able to work just as well across the border as for the system's native language. The demands for the comparison software increase along with the number of different languages.

For example: the Celtic name "Ewan" should be pronounced roughly as "you-in". A German employee who is not familiar with English might try to record the name in his own language as "ju-in". If this entry is later compared to the main office's records using German rules, any "duplication" will not be found. For a proper association, the comparison software must consider the relevant aspects for both languages.

As long as we are working only with Roman letters, then it is simply a matter of different rules for different languages. However, if new markets in Russia, India, or China are under consideration, then it becomes significantly more complex. These languages are written with completely different character sets and systems, using rules that are fundamentally at odds with western European languages. For example, Arabic is written from right to left, and vowels are omitted entirely.

The weakness of Unicode is in transcription

Unicode is an international standard, in which a digital code has been assigned ▶

Symbols worldwide

Alphabets - Алфавиты

In an alphabet, each letter corresponds to a sound. These are also referred to as phonographic scripts.

Examples of Alphabets:
Roman (Latin); Cyrillic; Greek

Abjads - العربية

Abjads consist exclusively of consonants. Vowels are omitted from most words, because they are obvious for native speakers, and are simply inserted when speaking. In addition, Abjads are normally written from right to left.

Examples of Abjads:
Hebrew; Arabic

Abugidas - वनागर

Abugidas are characteristic for scripts in India and Ethiopia. In this style, only the consonants are normally written, and standard vowels are assumed. If a different vowel is required, it is indicated with a special mark. Abugidas form an intermediate level between alphabetic and syllabic scripts.

Examples of Abugidas:
Hindi (Devanagari); Singhalese

Syllabic scripts -あいこ

Like alphabets, syllabic scripts are another type of phonographic script. In a syllabic script, each character stands for a syllable.

Examples of Syllabic Scripts:
Japanese (Hiragana, Katakana); Cherokee

Symbolic Scripts - 愛子

In symbolic scripts, each character is an ideogram standing for a complete word. Compound terms or concepts are composed of multiple symbols. Symbolic scripts are also called logographic scripts.

Examples of Symbolic Scripts:
Chinese; Japanese (Kanji)

for every meaningful script symbol and text element for every written language and symbolic system. A simple trick permits comparisons of international data. The first step is to assign a unique identification number to each data record. To compare German and Japanese addresses with one another, the Japanese symbols are converted (transliterated) into the Roman alphabet. The actual comparison then processes the data in Roman letters.

This approach seems promising at first, but closer examination quickly reveals its weaknesses. The conversion process itself destroys valuable information that is important for a "fuzzy" comparison.

The following sample comparison between English and Russian names demonstrates the weakness: "Fyodor" (a Russian name) is spelled in Cyrillic as "Федор". The Cyrillic letter "e" may be transliterated in Roman letters as "e" or as "ye". A diacritical symbol over the "e" results in "ë", and also changes the pronunciation to "yo". Unfortunately, this symbol may be omitted entirely in written Russian.

Using Unicode, the name Федор (Fyodor) would be converted to Fyedor or Fedor, depending on the set of rules used. A subsequent comparison using the Roman letters will run into problems, because the similarity between Fyodor and Fedor is relatively low. A reliable association between the names is no longer possible.

The conclusion: "Unicode" is not enough! Transcriptions and transliterations alone simply do not work, because the various script systems are based on different working principles. These principles must be respected as the data are compared.

Worldmatch® - Reliable and precise comparisons

Worldmatch is an algorithm that conquers international barriers. The advantage of this method is that individual scripts are not unified to a common character set. Instead, the differing scripts and character sets involved are compared directly.

Instead of transliterating, Worldmatch associates, increasing the precision of the comparisons enormously. Worldmatch checks the various scripts against each other, paying attention to the particularities of the languages involved, and recognizes similarities involved with typographical errors, exchanged characters, and abbreviations. This permits accurate international comparisons adhering to strict standards.

For our example using the Russian name "Fyodor" (Федор), this means that Worldmatch considers all of the possible Roman-letter spellings: Fedor, Fyedor, and Fyodor.

Worldmatch is available as part of Omikron Address-Center application for address management, and as a feature within Omikron Data Quality Server (for SOA). Integration in custom applications is also possible.

Written Form	Character Set
アイコ	Katakana
あいこ	Hiragana
あい子	Hiragana / Kanji
あ以子	Hiragana / Kanji
アイ子	Katakana / Kanji
あ衣子	Hiragana / Kanji
亜衣古	Kanji
亜伊子	Kanji
亜緯子	Kanji
亜以子	Kanji

Various possible written forms for the Japanese name „Aiko"

Chinese Names	
张爱国	ZHANG Aiguo
张爱民	ZHANG Aimin
张爱党	ZHANG Aidang

Zhang = Family Name, Ai = Generational name
In Chinese, the personal given name is represented by just the last syllable

Russian Names	
Михаил Горбачёв	Michail Gorbatschow
Раиса Горбачёва	Raissa Gorbatschowa

Arabic Names	
Father	Hassan ibn Selim
Son	Yassir ibn Hassan

In Arabic, the name of the father is part of each child's name.

Greek Names	
Πέτρος Κώτης	Petros Kotis
Αναστασία Κώτη	Anastasia Koti

The family name is adjusted to match gender in both Russian and in Greek

Omikron Data Quality GmbH
Pfaelzerstr. 35
75177 Pforzheim
Germany

Phone: +49-7231-12597-0
E-Mail: info@omikron.net
Web: www.omikron.net