Acquiring and Using Large Scale Knowledge

Michael Witbrock
Cycorp d.o.o.
30 Teslova Cesta, Ljubljana, SI-1000 Slovenia
witbrock@cycorp.eu

Formalized representations of knowledge can offer greater flexibility and broader applicability to problem solving programs, by reducing the need for specialized programming. However, this promise can only be fully realized if all of the knowledge required for some task is available in suitable form.

The recent increase in popularity of logic-based representations, operating over web-based logical symbols (the Semantic Web [ref]) offers some hope that the available knowledge might become available, but so far, the results of this representational activity have been mostly indicative, rather than broad demonstrations of the application of automated reasoning to answer real, practical needs.

One reason for this is that the knowledge being represented is still very close to data; relatively few predicates are used to connect large numbers of individual entities. The possibility of using arbitrarily many predicates offers more flexibility than using a relational database would, but this flexibility is barely exploited. Because the rules relating predicates are not generally expressed at all, and the links between classes are barely expressed, these relationships must still be imbedded in software written in a conventional programming language. To make things worse, the relationships that are expressed are often incorrect. The owl:sameAs relationship, for example, which is intended to express logical equivalence between symbols, is often used to connect terms that are merely closely related. At best, this sort of imprecision, which is probably unavoidable in logical systems built by human beings, is a substantial barrier to applying automated reasoning. At worst, it produces representations that are less useful than natural language text for machine use.

Realising the promise of automated reasoning will require building, perhaps on top of the current Semantic Web, the set of precise, interconnected representations of classes and individuals that are needed to permit the use of content-neutral reasoning systems (such as predicate calculus), and the set of rules and process descriptions that allow those reasoning systems to solve real problems and perform real tasks.

Human beings are no more adept at producing these systems than at producing the inflexible application specific computer programs we might hope to produce, but will have to be involved in the process. In the Cyc project, we are trying to produce human-computer collaborative systems that can build large scale knowledge bases that are both sufficiently general and sufficiently correct to be useable for solving problems that were not conceived when the knowledge bases were designed.

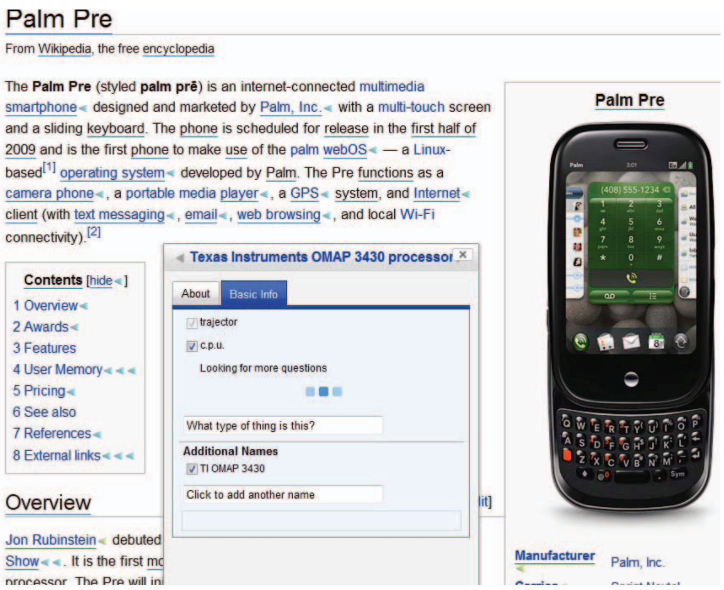
Broadly, the effort falls into three main parts: creating new, precise, logical terms for classes of objects and situations in the world, and for the objects and situations themselves; extending those representations with enough descriptive detail to support useful inferences about them; learning the generalizations that support those inferences (typically in the form of rules); and applying those inferences to new knowledge to validate both the knowledge, and the generalizations.

Acquiring sufficiently many new terms is a daunting prospect: manual knowledge entry in the first 25 years of the project has produced hundreds of thousands of interrelated logical terms, comparable to the vocabulary of a natural language. But the requirement for precision in terms used for formal inference suggests that a very much larger vocabulary is needed. If nothing else, the system will need terms for the many millions of commonly mentioned people, organizations, and events, which are in natural language represented using multiword phrases, for which precise rules for compositional semantics do not exist. The manually constructed terms, then, must be regarded as a seed. We are currently growing the term set around that seed, and acquiring associated knowledge, in two ways: collaborative, mixed initiative natural language interfaces, and automated reading.

To increase the number of terms, we are currently engaged in an effort to automatically acquire the most accessed concepts from Wikipedia, connected correctly to the Cyc ontology (see sw.opencyc.org e.g. http://sw.opencyc.org/concept/Mx4rvnNcIpwpEbGdrcN5Y29ycA to explore that ontology). That effort will be described in more detail elsewhere. Here we will concentrate on interactive KB augmentation.

Knowledge Capture: Mixed Initiative

Our main interface for collaborative knowledge acquisition is called the CURE (for Content Understanding, Review, or Entry), a dynamically constructed AJAX front-end for Cyc. Its name reflects our belief that knowledge capture should minimally intrusive: the CURE attempts to Understand elements of a document; if it is unsure, it elicits human Review; as a back-off, it supports mixed initiative natural language Entry. In the screen shots below, the CURE is shown gathering information from Wikipedia pages that have been automatically tagged with OpenCyc semantic terms (in the case of blue underlines) and unknown named entities, in the case of green underlines. Two examples are given. In one, the TI OMAP 3430 processor used in the palm pre is created as a concept, and identified as a type of processor.


Palm Pre

From Wikipedia, the free encyclopedia

The **Palm Pre** (styled **palm prē**) is an internet-connected multimedia smartphone designed and marketed by Palm, Inc. with a multi-touch screen and a sliding keyboard. The phone is scheduled for release in the first half of 2009 and is the first phone to make use of the palm webOS — a Linux-based[1] operating system developed by Palm. The Pre functions as a camera phone, a portable media player, a GPS system, and Internet client (with text messaging, email, web browsing, and local Wi-Fi connectivity).[2]

Contents [hide]
1 Overview
2 Awards
3 Features
4 User Memory
5 Pricing
6 See also
7 References
8 External links

Overview

Jon Rubinstein debuted
Show. It is the first mo
processor. The Pre will in

Texas Instruments OMAP 3430 processor

About | Basic Info

☐ trajector
☑ c.p.u.

Looking for more questions

What type of thing is this?

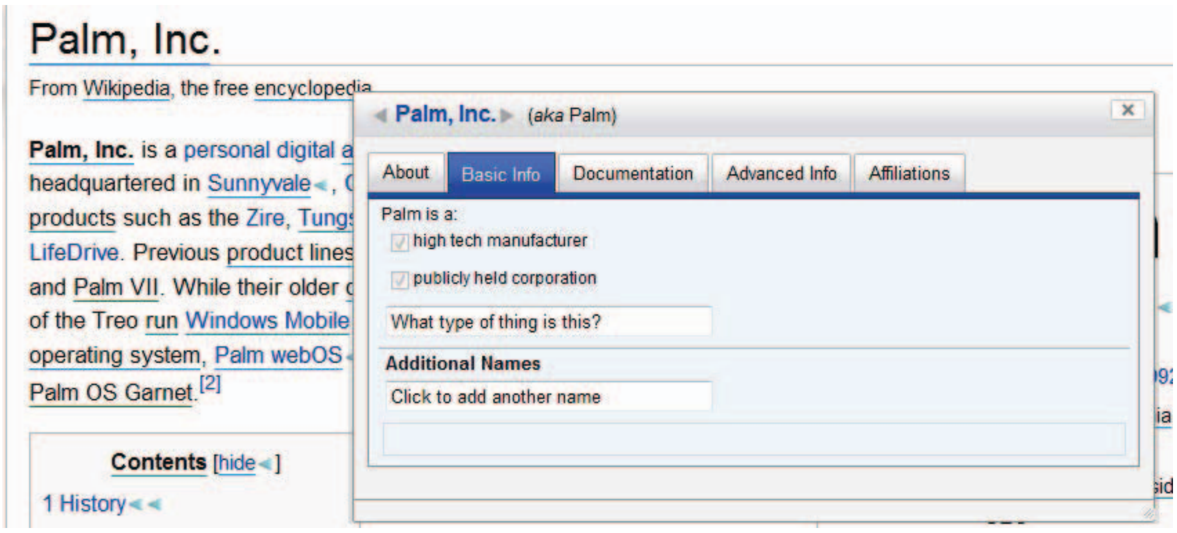
Additional Names
☑ TI OMAP 3430

Click to add another name

Palm Pre

Manufacturer: Palm, Inc.

In the other, below, additional information about the existing concept, Palm, Inc is added, using forms generated based on the existing background knowledge that Palm Inc is a high-tech manufacturer and a public company.


Palm, Inc.

From Wikipedia, the free encyclopedia

Palm, Inc. is a personal digital a
headquartered in Sunnyvale, C
products such as the Zire, Tungs
LifeDrive. Previous product lines
and Palm VII. While their older c
of the Treo run Windows Mobile
operating system, Palm webOS
Palm OS Garnet.[2]

Contents [hide]
1 History

Palm, Inc. (aka Palm)

About | Basic Info | Documentation | Advanced Info | Affiliations

Palm is a:
☐ high tech manufacturer
☐ publicly held corporation

What type of thing is this?

Additional Names

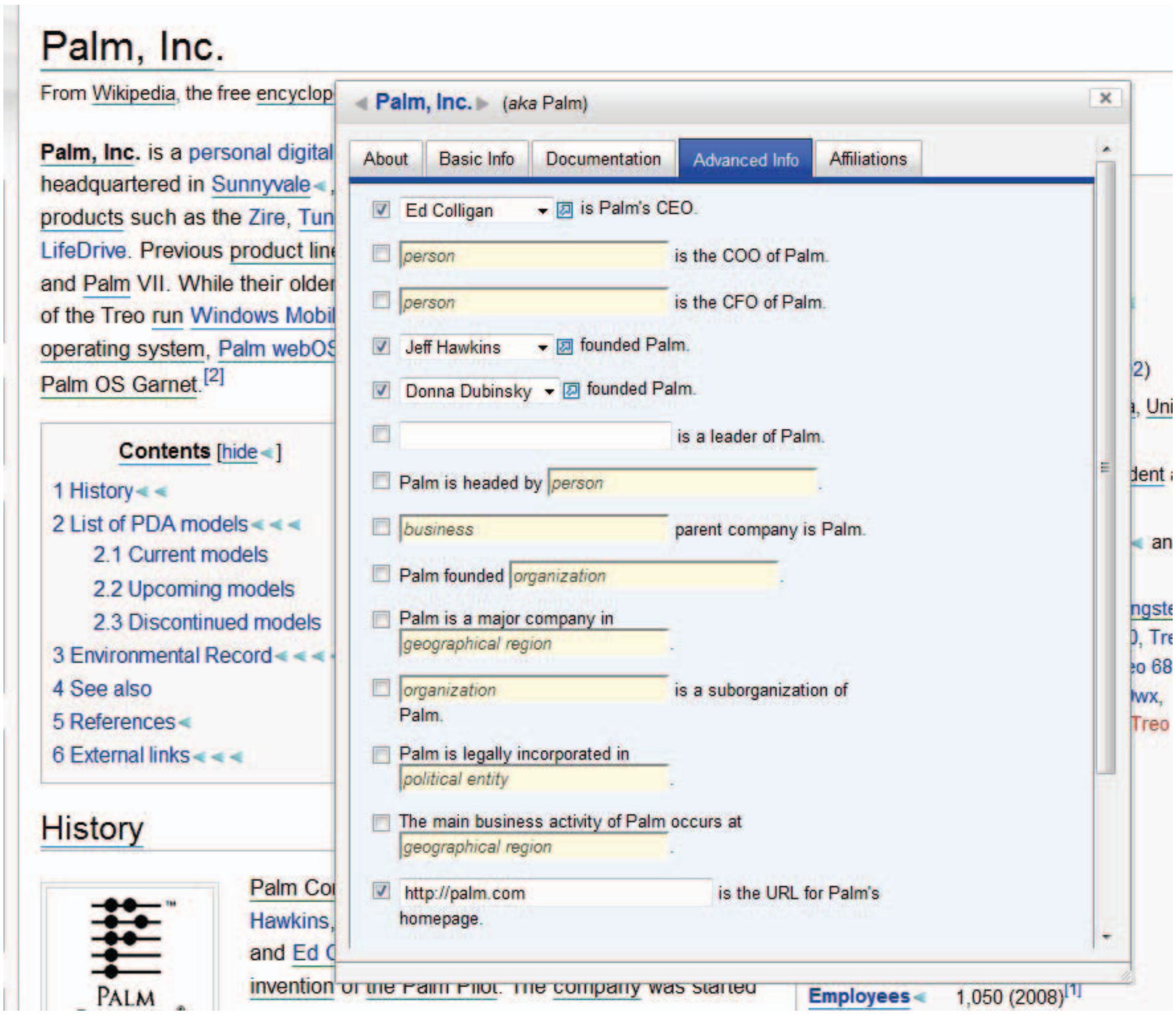
Click to add another name

These forms, a more complex example of which is shown below, are not pre-determined, they are generated from the background knowledge base using inference. For example the KB content:

(generateFormulasForElements-TermIsa PubliclyHeldCorporation
    (TheSet stockTickerSymbol))
(generateFormulasForElements-TermIsa CommercialOrganization
    (TheSet importantCompany
     subOrganizations
     mainBusinessActivityOfOrgOccursAt
        foundingAgent incorporatedIn))

both apply to PalmInc (because (isa PalmInc PublicalyHeldCorporation) is true, as is (isa PalmInc CommericalOrganizarion) by transitivity, and tell the system some of the relations it should seek to instantiate for Palm. The remaining work of assembling the forms an generating the NL content for them is done automatically, based on KB content attached to the predicates in question. Since the goals for knowledge acquisition are stored in the KB, this way, they themselves can be produced by inference, on the basis of knowledge accumulating in the KB (via rules that conclude to (generatedFormulasForElements-TermIsa ?X ?Y))

The information entered by the user is added to the knowledge base, and can be reproduced using natural language, as shown below, or used widely for query answering.



Palm, Inc.

From Wikipedia, the free encyclop

Palm, Inc. is a personal digital headquartered in Sunnyvale, products such as the Zire, Tun LifeDrive. Previous product lin and Palm VII. While their older of the Treo run Windows Mobi operating system, Palm webOS Palm OS Garnet.[2]

Contents [hide]
1 History
2 List of PDA models
2.1 Current models
2.2 Upcoming models
2.3 Discontinued models
3 Environmental Record
4 See also
5 References
6 External links

History

Palm Co Hawkins, and Ed C invention of the Palm Pilot. The company was started

Employees 1,050 (2008)[1]

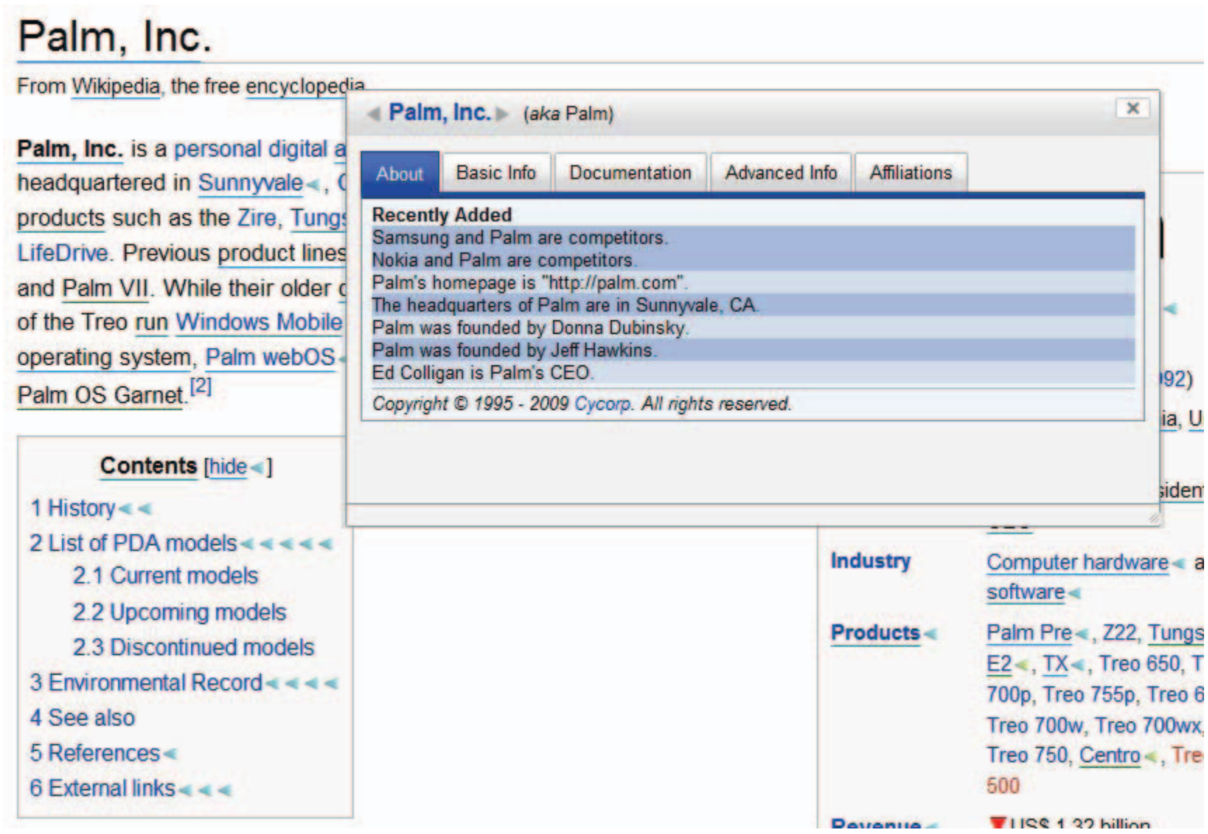
Palm, Inc. (aka Palm)

About | Basic Info | Documentation | Advanced Info | Affiliations

✓	Ed Colligan is Palm's CEO.
☐	person is the COO of Palm.
☐	person is the CFO of Palm.
✓	Jeff Hawkins founded Palm.
✓	Donna Dubinsky founded Palm.
☐	is a leader of Palm.
☐	Palm is headed by person.
☐	business parent company is Palm.
☐	Palm founded organization.
☐	Palm is a major company in geographical region.
☐	organization is a suborganization of Palm.
☐	Palm is legally incorporated in political entity.
☐	The main business activity of Palm occurs at geographical region.
✓	http://palm.com is the URL for Palm's homepage.

Producing More-General Rules from Ground Facts

In order to realize the potential of automated reasoning, it is necessary to move beyond simply looking up facts, or doing simple type generalizations. In previous work (Cabral et al. 2005) we described the use of ILP to produce candidate rules in Cyc. Recently, we have been experimenting with an even simpler technique: transforming decision trees trained on KB content (in this case from the Analyst KB section of the Cyc KB, Deaton et al. 2005) into Cyc rules with probabilistic conclusions.


Palm, Inc.

From Wikipedia, the free encyclopedia

Palm, Inc. is a personal digital a
headquartered in Sunnyvale, C
products such as the Zire, Tungs
LifeDrive. Previous product lines
and Palm VII. While their older c
of the Treo run Windows Mobile
operating system, Palm webOS
Palm OS Garnet.[2]

Contents [hide]
1 History
2 List of PDA models
2.1 Current models
2.2 Upcoming models
2.3 Discontinued models
3 Environmental Record
4 See also
5 References
6 External links

Palm, Inc. (aka Palm)

About | Basic Info | Documentation | Advanced Info | Affiliations

Recently Added
Samsung and Palm are competitors.
Nokia and Palm are competitors.
Palm's homepage is "http://palm.com".
The headquarters of Palm are in Sunnyvale, CA.
Palm was founded by Donna Dubinsky.
Palm was founded by Jeff Hawkins.
Ed Colligan is Palm's CEO.
Copyright © 1995 - 2009 Cycorp. All rights reserved.

Industry	Computer hardware a software
Products	Palm Pre, Z22, Tungs E2, TX, Treo 650, T 700p, Treo 755p, Treo 6 Treo 700w, Treo 700wx, Treo 750, Centro, Tre 500

For example, the decision tree that decides whether or not the organization Iraqi insurgents group is a likely perpetrator of an attack is simply this:

```
where = Iraq
|  when <= 2004.0
|  |  when <= 1998.0: other (3.0)
|  |  when > 1998.0: Iraqi-insurgents-Group (2.0)
|  when > 2004.0: Iraqi-insurgents-Group (79.0)
where != Iraq: other (1558.0)
```

The translation process is straightforward. Each decision tree is translated by incorporating rules only for leaf nodes with 30 or more corresponding instances, and which positively conclude to the perpetrator in focus. After filtering out rules failing to meet these conditions, only one rule was left in the case of the Iraqi insurgents group:

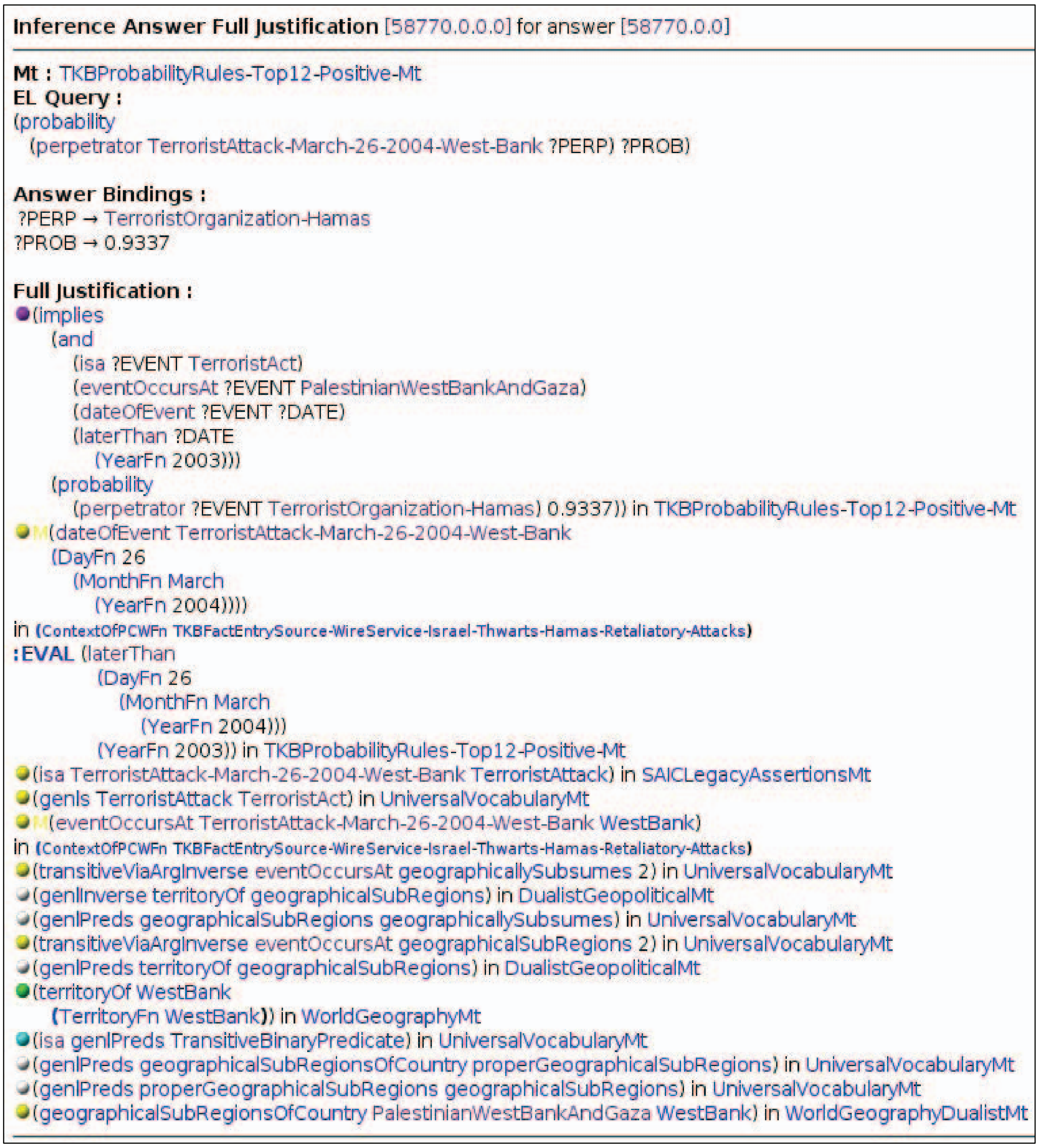
```
(implies
 (and
  (isa ?EVENT TerroristAct)
  (eventOccursAt ?EVENT Iraq))
 (probability (perpetrator ?EVENT Iraqi-insurgents-Group)
             0.9655))
```

Once entered into the KB, the learned rules, including this one, can be used to answer queries about likely perpetrators. For example, using other rules learned from the Analyst's Knowledge Base, one can ask the following query in order to find out the most likely perpetrators of the terrorist attack in the West Bank on March 26th, 2004.

```
(probability
 (perpetrator TerroristAttack-March-26-2004-West-Bank ?PERP) ?PROB)
```

One of the bindings returned is TerroristOrganization-Hamas, who according to learned rules, committed the act with 93%

likelihood according to Cyc. This conclusion depends on learned rules, ground facts, and general "common-sense" knowledge. In order to understand how that conclusion was reached, it is useful to view the justification for that answer, which gives the facts and rules that were used to conclude it. The justification is shown in the screenshot below.



```
Inference Answer Full Justification [58770.0.0.0] for answer [58770.0.0]

Mt : TKBProbabilityRules-Top12-Positive-Mt
EL Query :
(probability
  (perpetrator TerroristAttack-March-26-2004-West-Bank ?PERP) ?PROB)

Answer Bindings :
 ?PERP → TerroristOrganization-Hamas
?PROB → 0.9337

Full Justification :
●(implies
    (and
      (isa ?EVENT TerroristAct)
      (eventOccursAt ?EVENT PalestinianWestBankAndGaza)
      (dateOfEvent ?EVENT ?DATE)
      (laterThan ?DATE
        (YearFn 2003)))
    (probability
      (perpetrator ?EVENT TerroristOrganization-Hamas) 0.9337)) in TKBProbabilityRules-Top12-Positive-Mt
●M(dateOfEvent TerroristAttack-March-26-2004-West-Bank
    (DayFn 26
      (MonthFn March
        (YearFn 2004))))
in (ContextOfPCWFn TKBFactEntrySource-WireService-Israel-Thwarts-Hamas-Retaliatory-Attacks)
:EVAL (laterThan
          (DayFn 26
            (MonthFn March
              (YearFn 2004)))
          (YearFn 2003)) in TKBProbabilityRules-Top12-Positive-Mt
●(isa TerroristAttack-March-26-2004-West-Bank TerroristAttack) in SAICLegacyAssertionsMt
●(genls TerroristAttack TerroristAct) in UniversalVocabularyMt
●M(eventOccursAt TerroristAttack-March-26-2004-West-Bank WestBank)
in (ContextOfPCWFn TKBFactEntrySource-WireService-Israel-Thwarts-Hamas-Retaliatory-Attacks)
●(transitiveViaArgInverse eventOccursAt geographicallySubsumes 2) in UniversalVocabularyMt
●(genlInverse territoryOf geographicalSubRegions) in DualistGeopoliticalMt
●(genlPreds geographicalSubRegions geographicallySubsumes) in UniversalVocabularyMt
●(transitiveViaArgInverse eventOccursAt geographicalSubRegions 2) in UniversalVocabularyMt
●(genlPreds territoryOf geographicalSubRegions) in DualistGeopoliticalMt
●(territoryOf WestBank
    (TerritoryFn WestBank)) in WorldGeographyMt
●(isa genlPreds TransitiveBinaryPredicate) in UniversalVocabularyMt
●(genlPreds geographicalSubRegionsOfCountry properGeographicalSubRegions) in UniversalVocabularyMt
●(genlPreds properGeographicalSubRegions geographicalSubRegions) in UniversalVocabularyMt
●(geographicalSubRegionsOfCountry PalestinianWestBankAndGaza WestBank) in WorldGeographyDualistMt
```

Scaling Inference

A third component of the effort to support Artificial Intelligence at scale is the ability to do relatively complex inference such as the one shown above at extremely large scale. As part of its goal of supporting reasoning over all the world's knowledge, Cycorp Europe is involved in the LarKC FP7 IP project. The LarKC platform, based in large part on the Cyc inference engine and the Ontotext OWLIM triple store, aims to support heterogeneous reasoning over billions of billions of assertions.

LarKC (The Large Knowledge Collider) is a platform for massive distributed incomplete reasoning that aims to remove the scalability barriers of currently existing reasoning systems for the Semantic Web. LarKC users compose plug-ins, implementing specific elements of problem solving, into dynamically reconfigurable problem solving workflows. Component plug-ins may perform data selection (for example, finding RDF triples on the web that might be of interest to a particular user), data transformation (for example extracting assertions about gene expression from text), reasoning (logical, or probabilistic), etc. A key feature of the LarKC design is the exploitation of the current evolution towards very large scale parallelism (multi-core and massively multi-core processors), LarKC is currently investigating how to apply distributed computing and traditional high performance computing techniques (such as OpenMP and MPI) within data access and reasoning plug-ins that are naturally fine-grained, and across plug-ins that are naturally coarse grained. As an open platform, LarKC encourages use by external and developers, making it possible to achieve a higher degree of sophistication through the combination of their own techniques with the ones developed within the LarKC Community (by both LarKC developers and other external adopters).

The sort of scalability provided by LarKC-developed techniques will support the use of more data in serving user information needs, certainly, but can also contribute to the development of very large scale knowledge-based systems, by supporting scaling of knowledge acquisition, and introspection, including rule induction, for knowledge refinement.

References

[1] Cabral, J., Kahlert, R. C., Matuszek, C., Witbrock, M., & Summers, B. (2005). Converting Semantic Meta-knowledge into Inductive Bias . In Inductive Logic Programming (pp. 38-50). Berlin: Springer.

[2] Deaton, C., Shepard, B., Klein, C., Mayans, C., Summers, B., Brusseau, A., Witbrock, M. Lenat, D. (2005). The Comprehensive Terrorism Knowledge Base in Cyc. In Proceedings of the 2005 International Conference on Intelligence Analysis. McLean, Virginia.

[3] Panton, K., Miraglia, P., Salay, N., Kahlert, R, Baxter, D, Reagan, R. (2002) Knowledge Formation and Dialogue Using the KRAKEN Toolset In Proceedings of the Fourteenth National Conference on Innovative Applications of Artificial Intelligence, p. 900-905. Edmonton, Canada.