Insurance Portfolio Management

Michael Genesereth
Computer Science Department
Stanford University

Technical Collaborators:
Preston Carlson, Enci Liu, Sophia Ramsay

In this brief presentation, I am going to talk one particular problem we are addressing in our research, viz. Insurance Portfolio Management. This problem is interesting for two reasons. (1) First of all, it is a perennial problem faced by insurance consumers. (2) Second, it is a good illustration of the power of computable contracts technology - the problem can be solved using computable contracts technology, but it cannot reasonably be solved using other approaches to automating insurance processes.

The presentation begins with a brief introduction to insurance portfolio management. This is followed by a demonstration of a computer system that supports insurance portfolio management. The presentation then concludes with a look at the underlying technology used in the demonstration.

As insurance consumers, we often think of insurance policies as being partitioned into distinct areas – home insurance, auto insurance, travel insurance, credit card insurance, and so forth; and we frequently buy different policies to provide coverage in these different areas.

In reality, things are more complicated, with policies in different areas often providing overlapping coverage. If we are unaware of these overlaps, we can end up paying more for insurance than we need; on the flip side, there can be coverage gaps of which we are unaware.

Example

For example, rental car damage may be covered by a policy purchased from a personal auto insurance policy, a travel insurance policy, in some cases a homeowners policy, and a credit card, which raises the question of whether we need to purchase collision insurance when we rent from Hertz. If we use a major credit card to book the rental, the answer is often no. However, there are exceptions. For example, when we are traveling in Ireland, it is sometimes necessary to purchase additional insurance since credit card insurance often does not apply there.

Insurance Portfolio Management

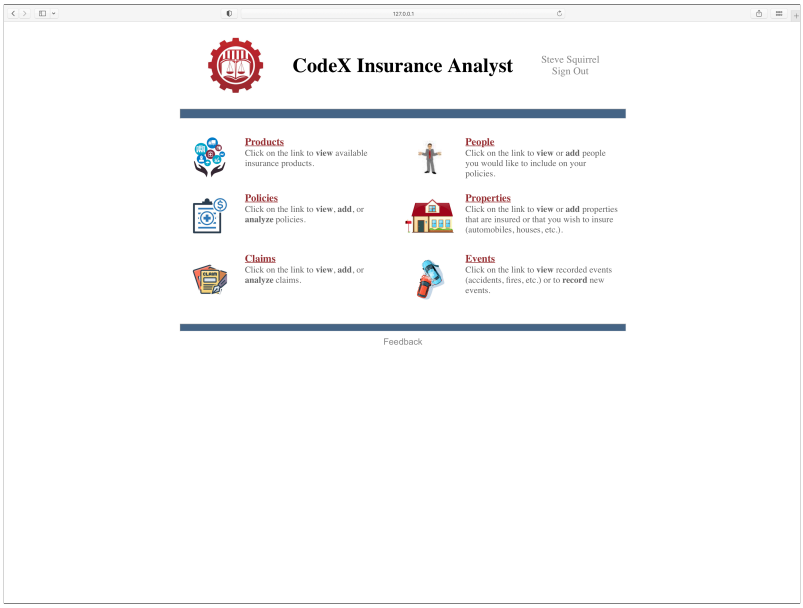
Insurance portfolio management (IPM) is the process of
managing *overlapping insurance policies*
with an eye to *minimizing insurance costs*
while *ensuring adequate coverage*.

Making decisions like these is an example of Insurance portfolio management. In our work, we define Insurance portfolio management (IPM) to be the process of managing multiple, potentially overlapping insurance policies, with an eye to minimizing insurance costs while ensuring adequate coverage.

Complexity of Insurance Policies

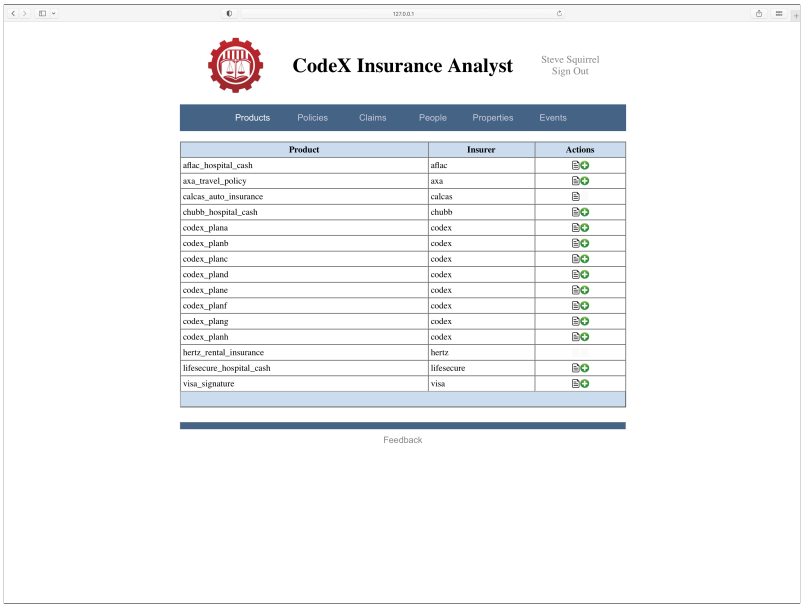
Policies are complex, incomplete, inconsistent.

Of course, we can do this ourselves. By studying the contracts associated with insurance policies, we can avoid overlaps and ensure there are no gaps in coverage. The problem is that insurance portfolio management is not easy. Insurees usually do not have the time or patience to compare policies from multiple insurance providers; and, even if they have the time, they often do not have the legal background needed to understand the complex legal wording of the lengthy, 100-page contracts associated with those policies.

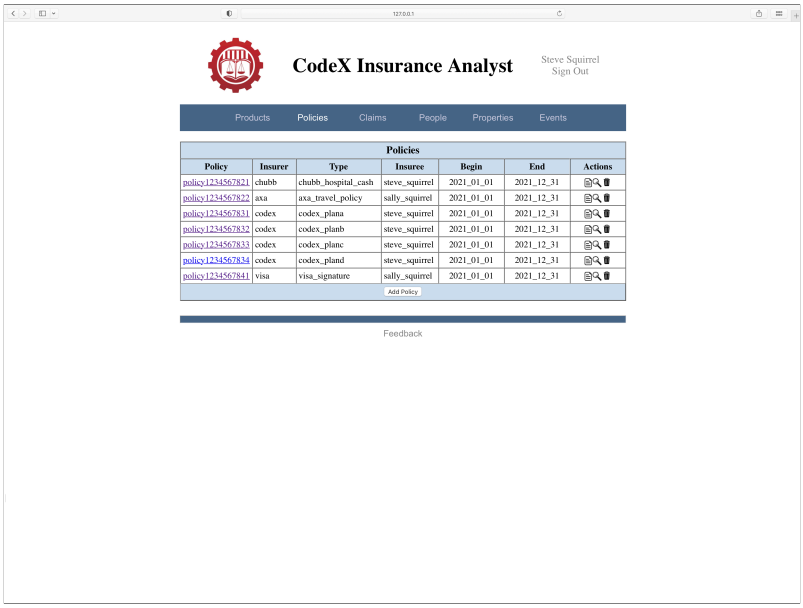


In order to deal with this problem, we have been investigating ways to automate the process of insurance portfolio management. The current manifestation of this work a web-based application called the CodeX Insurance Analyst (CIA). CIA incorporates data and policies from multiple insurance companies, and it provides users with analysis across these policies. We are currently focussed on helping individuals. In the future, we expect to provide similar capabilities for businesses and insurance companies.

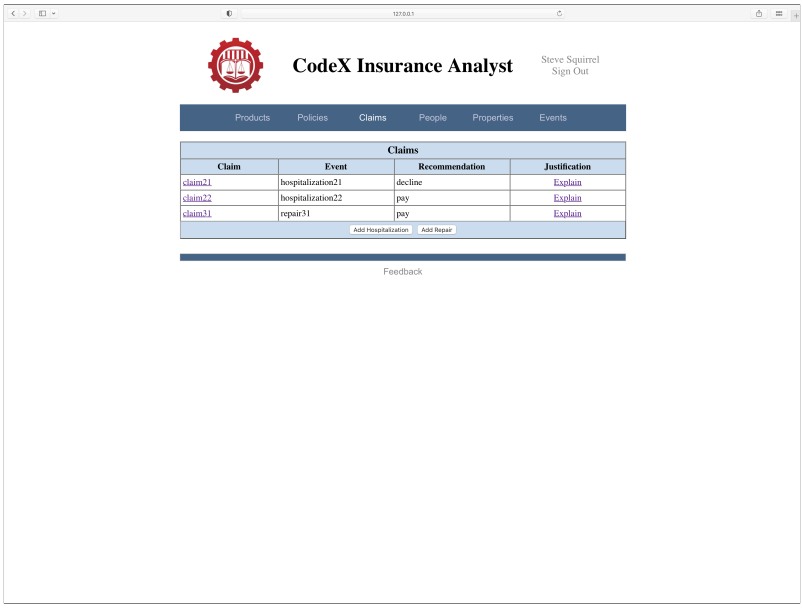
Let's take a brief tour of the current prototype. The first page one sees after signing in to CIA is the splash page for the system. The tabs here indicate the principals types of entities represented in the system - insurance products, policies, claims, people, properties (such as vehicles or houses), and events (such as accidents and hospitalizations). Note that we are logged in as a particular customer named Steve Squirrel.



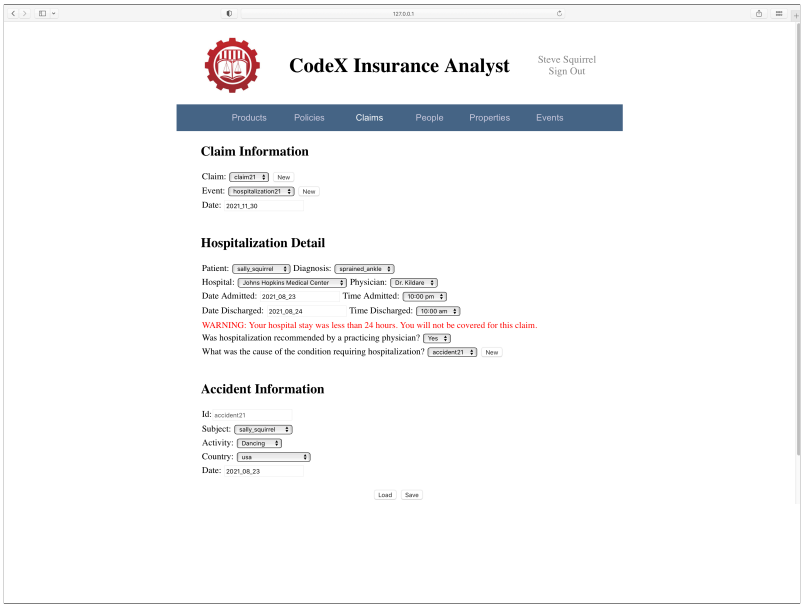
The products page contains a list of insurance products available in the demonstration system. At presen we have only a dozen or so in our system. Clicking on the text icon shows a textual description of a policy. It is possible to specify a product for analysis by clicking the + button.



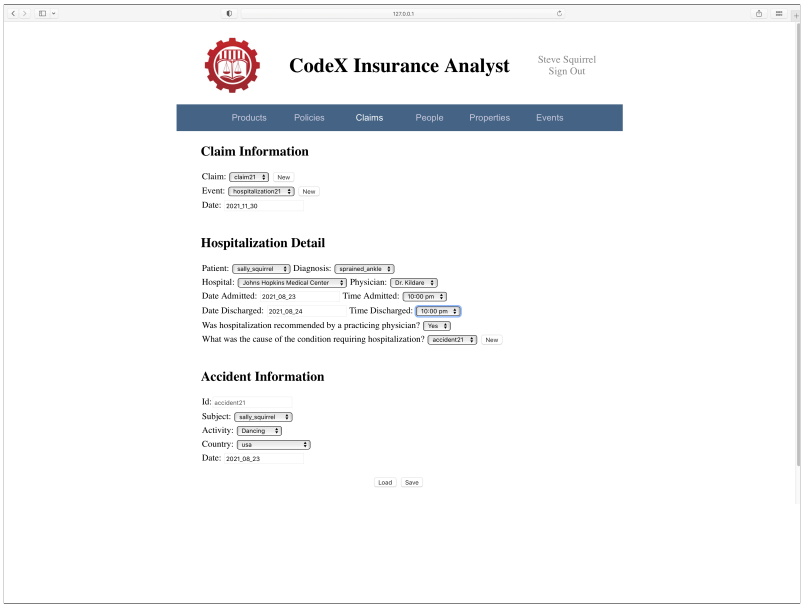
The Policies page shows the insurance policies the user has selected for analysis. A policy is essentially an insurance product, specialized to a particular person, for a specific range of dates, with specific riders, and so forth. Again, clicking on a text icon shows a product description. Clicking on a magnifier requests a comparative analysis of that policy. More on this later. Clicking the trash can deletes a policy.



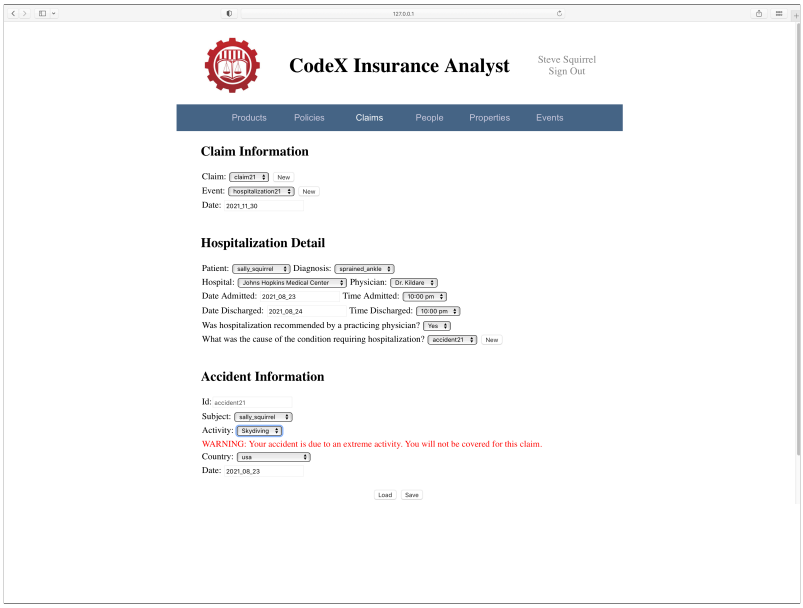
The Claims page shows real or hypothetical claims the user would like to analyze. In Steve's case, there are two hospitalization claims and one vehicle repair claim already specified. Given the policies Steve has selected, one of the hospitalization claims should be declined, while the other should be paid. Clicking the explanation button next to a claim gives an explanation of the listed recommendation. Additional claims can be added by clicking one of the Add buttons.



In order to see how claim analysis works, let's look at one of these claims. Clicking on the identifier for a claim brings up an interactive worksheet with details about that claim. The panel at the top provides overall information about the claim. The panel in the middle shows details of the relevant hospitalization. In this case, Steve's daughter Sally was treated for a sprained ankle at Johns Hopkins. At the bottom, is information about the event that led to the injury, in this case a mishap on the dance floor. In this case, the claim is denied because Sally's stay at the hospital was too short - just 12 hours.



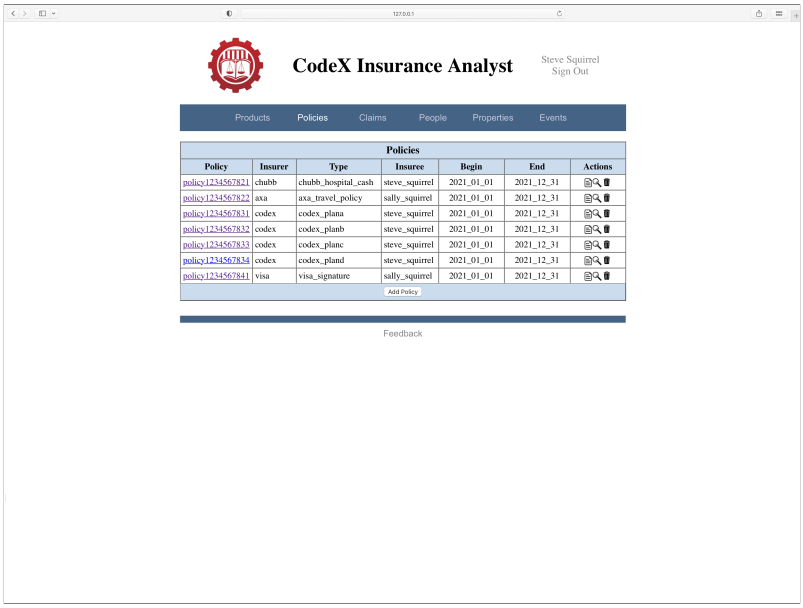
If we adjust the length of the stay, we notice that the claim is acceptable.



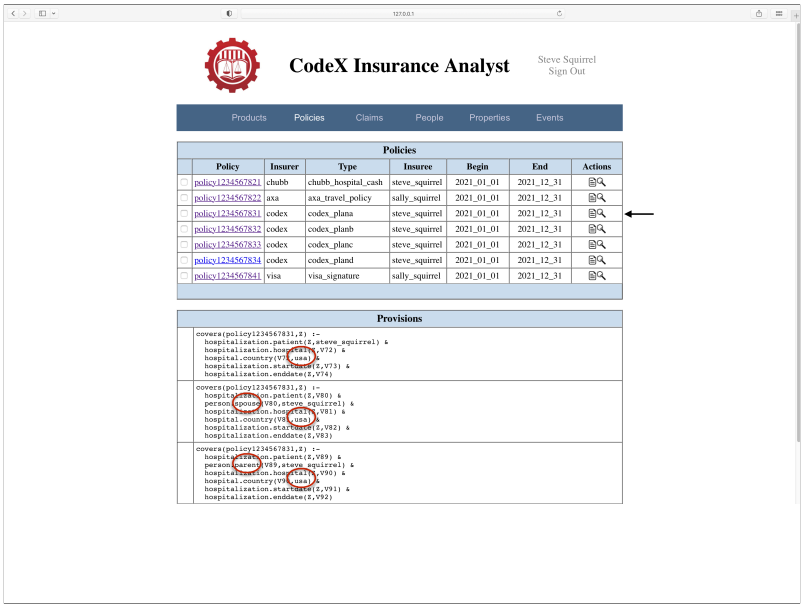
However, if Sally's sprained ankle was the result of skydiving rather than dancing, the claim would once again be declined since that is a hazardous activity that is explicitly excluded.

The main feature to note here is the transparency provided by the interactive worksheet. The rules defining the terms and conditions of the customer's policies are embedded in the worksheet and that allows the worksheet to give the claimant immediate feedback on whether the claim should be honored. Changing various values changes the prediction. Moreover, the worksheet is provides explanations for its conclusions so the claimant understands the reasons for its conclusions.

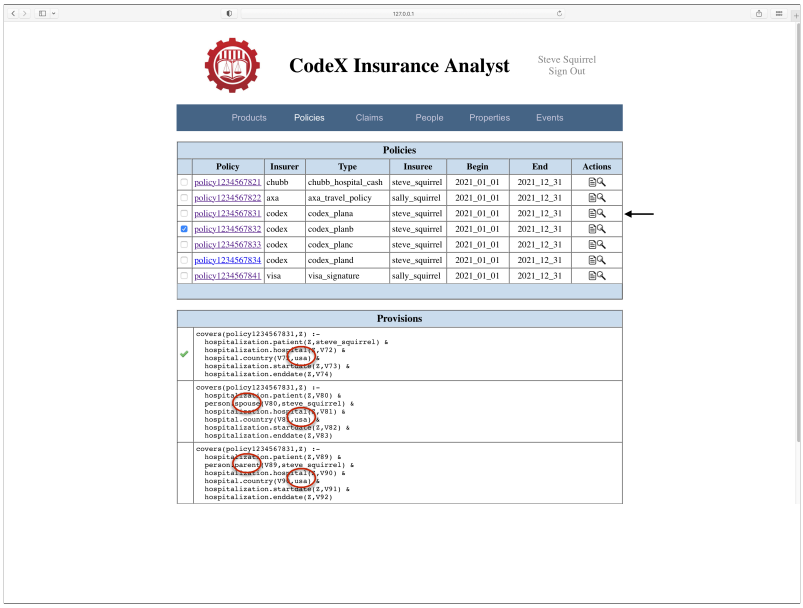
Note that the analysis here differs from traditional claim analysis in that the claim is analyzed against Steve's entire portfolio of policies, not just one policy. However, portfolio management is more general than this. It allows the user to analyze whole policies, not just specific claims. Let's see how this works.



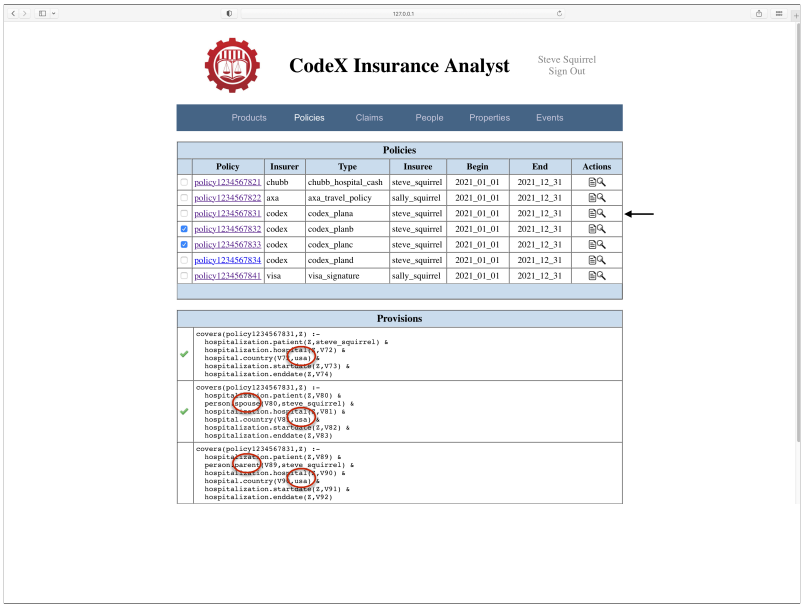
Clicking on the Policies tab takes us back to the policies page. For the purposes of the demo, we will focus on four fictional policies provided by Codex. Plan a covers Steve, his spouse, and children in the US. Plan b covers Steve anywhere in the world. Plan c covers his wife anywhere in the world. Plan d covers his children anywhere in the world.



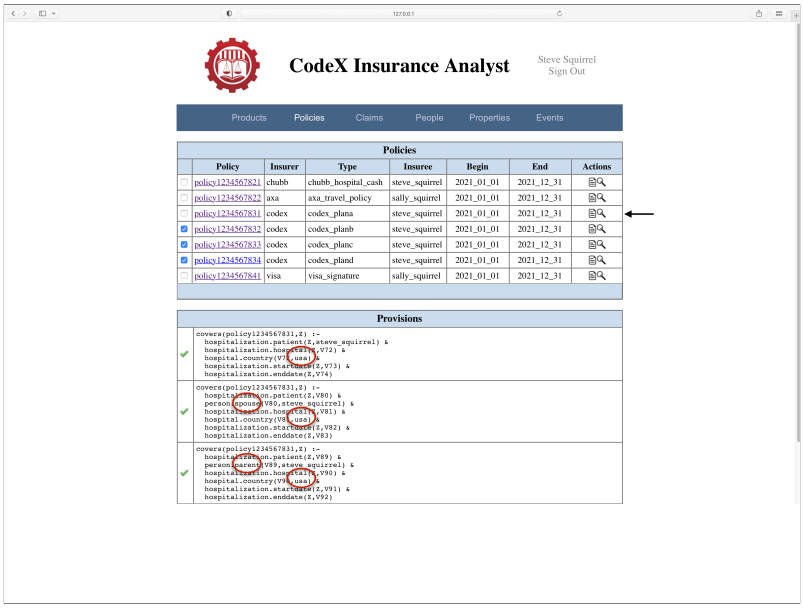
Clicking on the magnifier next to codex_plana policy shows the provisions of the policy - steve, spouse, and kids covered for hospitalizations in a US hospital. The details here are shown in our internal language. In the actual system, these would be rendered into natural language.



Recall that plan b covers steve anywhere in the world. Checking the planb box leads to a checkmark next to the first provision. On other words, plan b covers the first provision of plan a.

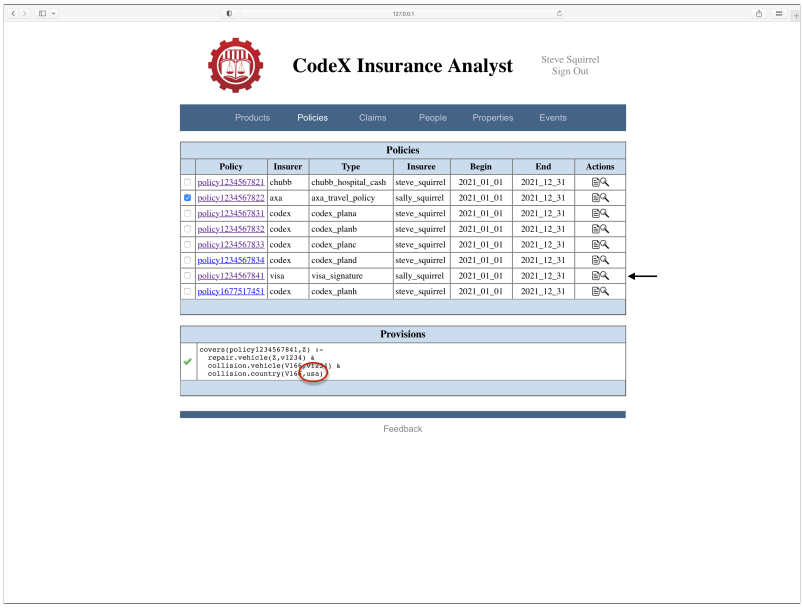


Planc covers his spouse anywhere in the world. Checking the plan c box leads to a checkmark next to the second provision. On other words, plan b covers the second provision.



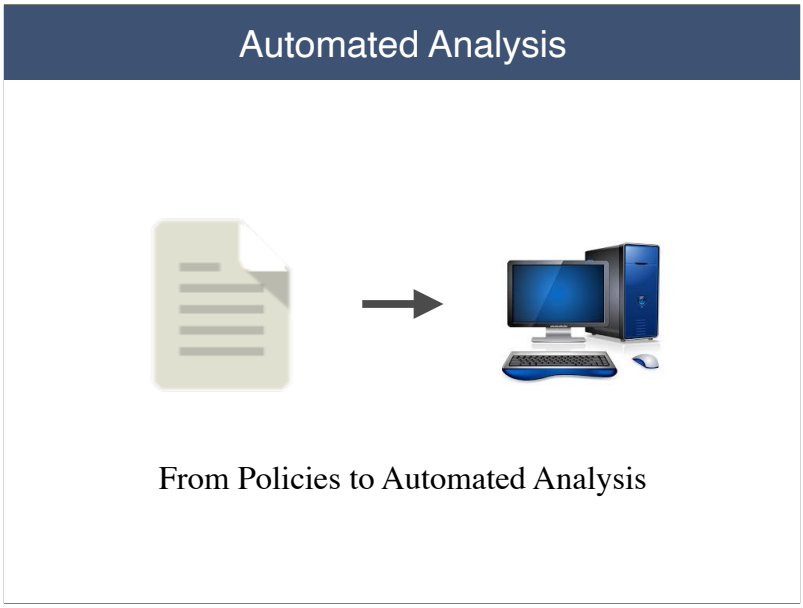
Plan d covers his kids. Checking the plan d box leads to a checkmark next to the third provision. In other words, plan c covers the first provision. The upshot is that these three policies taken together entirely cover the provisions of plan a.

Of course, if we were to reverse the analysis, pressing the magnifier next to plan b and checking the plan a box, nothing would happen in this case, since plan a covers only US hospitalizations, whereas plan b covers hospitalizations anywhere in the world.



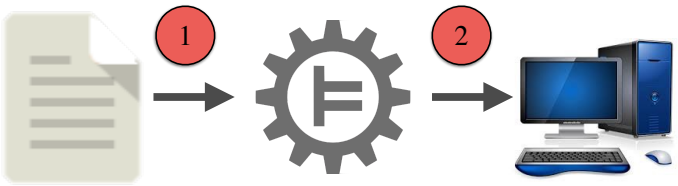
One more quick example, this time with real policies. The visa_signature policy provides coverage in the US, while AXA is world-wide. If we ask the system to analyze the visa-signature policy and check the axa-travel_insurance policy, we see that the AXA policy subsumes the visa_signature policy but not vice versa.

In general, coverage analysis is more complicated than these simple examples illustrate, e.g. when there are constraints on types of coverage, when there are overlapping policy dates, and when there are multiple interacting claims. And, of course, this demo focusses only on coverage. Payment analysis must also be done to deal with deductibles, overall limits on payment, and so forth. But, hopefully these examples illustrate the basic idea of the portfolio analysis. Time now to say a few words about the technology.



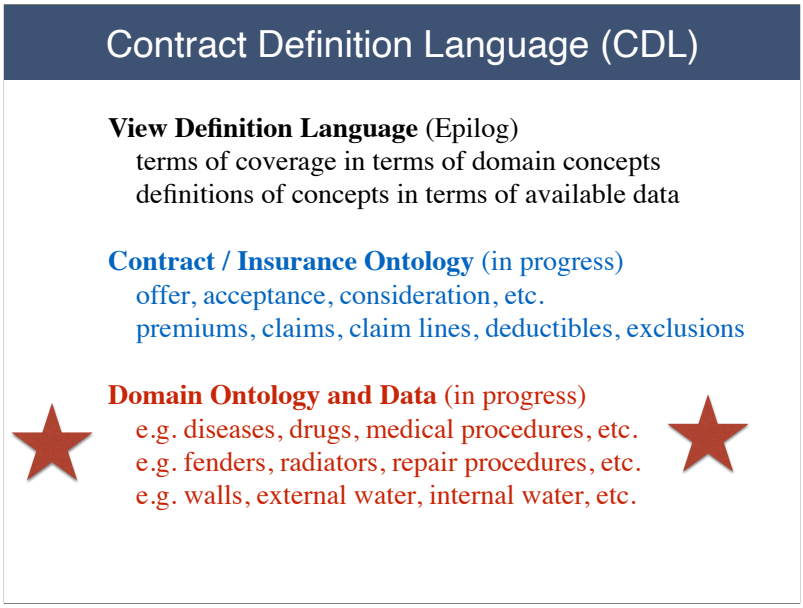
In order to create a portfolio analyst, we need to supply it with information about relevant insurance policies so that it can do its work.

Computable Contracts



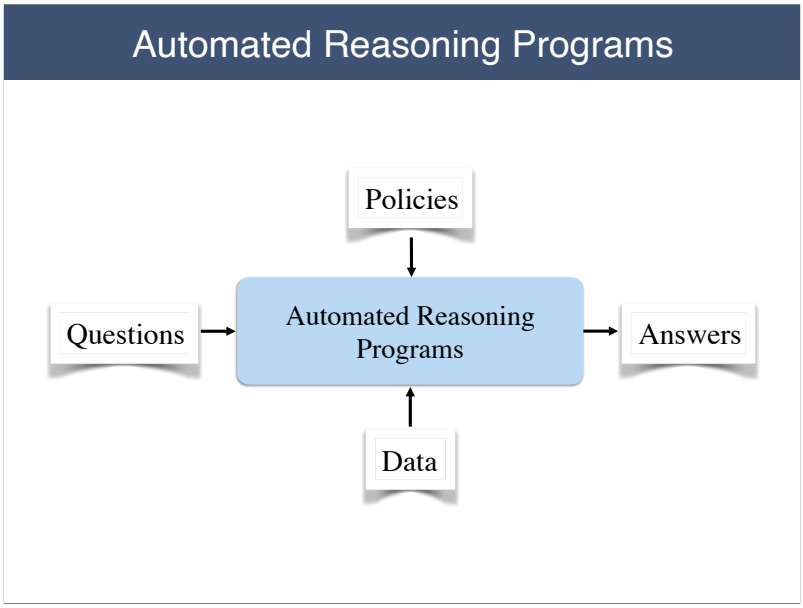
From Text to Computable Contracts to Automation

Our approach to this is based on Computable Contracts technology. (1) The first step is to encode polices as "computable contracts", i.e. formal representations of the terms and conditions of those contracts. (2) The second step is to supply these computable contracts as data to automated reasoning programs that can perform the desired analysis - thereby bringing the power of computers to provide the capabilities we just saw.



The key to formalizing polices is a contract definition language (CDL). There are three parts to the language we use - (1) a *view definition* language based on symbolic logic, (2) a *contract ontology*, and (3) a *domain ontology*. (The view definition language is loosely analogous to the grammar of English; the contract ontology consists of words for legal concepts; and the domain ontology consists of words for the subject matter or contracts.)

Of these three, the domain ontology requires the most work. Luckily, there are emerging standards in a variety of areas. For example, in the medical arena, there are formal classifications of diseases and drugs and procedures; and there are data bases knowledge bases defining these concepts in terms of each other, e.g. units and currency conversions.



Luckily, we know how to build automated reasoning programs that can reliably answer questions from data and rules encoded in CDL, and we can use these programs for multiple purposes - whether analyzing individual claims or analyzing whole policies.

Interactive Development Environment (IDE)

Finally, we are developing tools to facilitate the authoring, revision, and debugging of computable policies in CDL and in a form that ensures they can be used by the reasoning programs we are using. The computer programming community has long recognized the value of Interactive Development Environments (IDEs) to help in developing and maintaining programs. The use of a logical language like CDL makes possible the development of especially powerful IDEs for authoring, analyzing, and debugging policies as well.

Contractnet

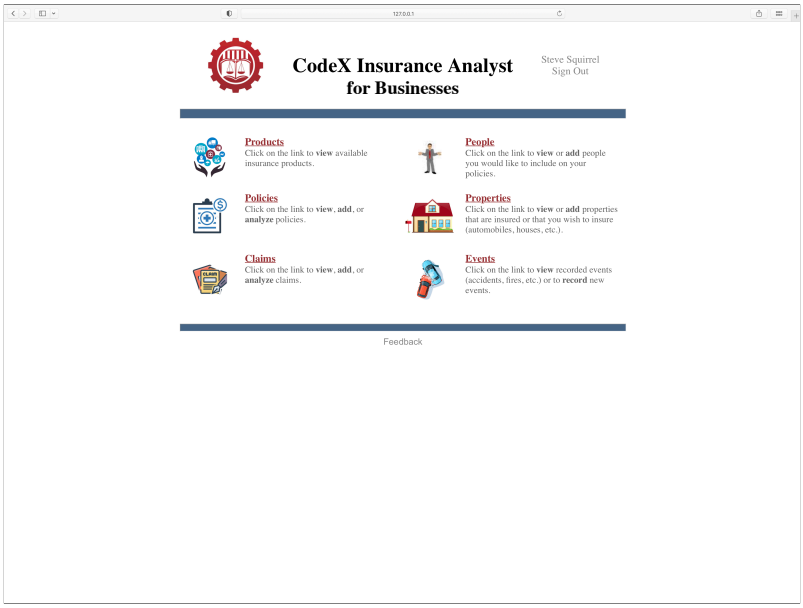
Background Datasets
- Geography
- Hospitals

Background Rulesets
- e.g. a week is 7 days
- e.g. parents are older than their children

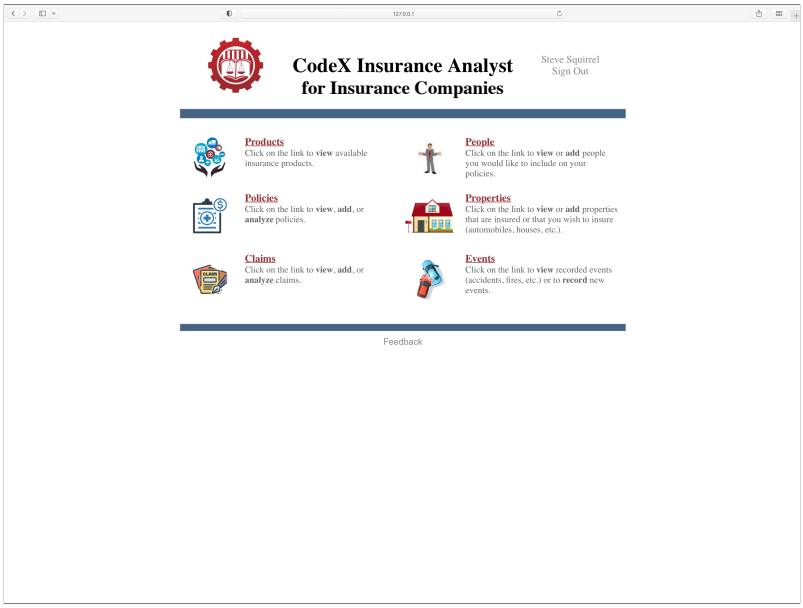
Open-Source Contract Processing Software
- Compliance checking (EpilogJS)
- Explaining contractual analysis (ExplainJS, EnglishJS)
- User Worksheets (WorksheetsJS)
- Containment Testing and Configuration Aids (CheckJS)

Github for Computable Insurance Contract Software

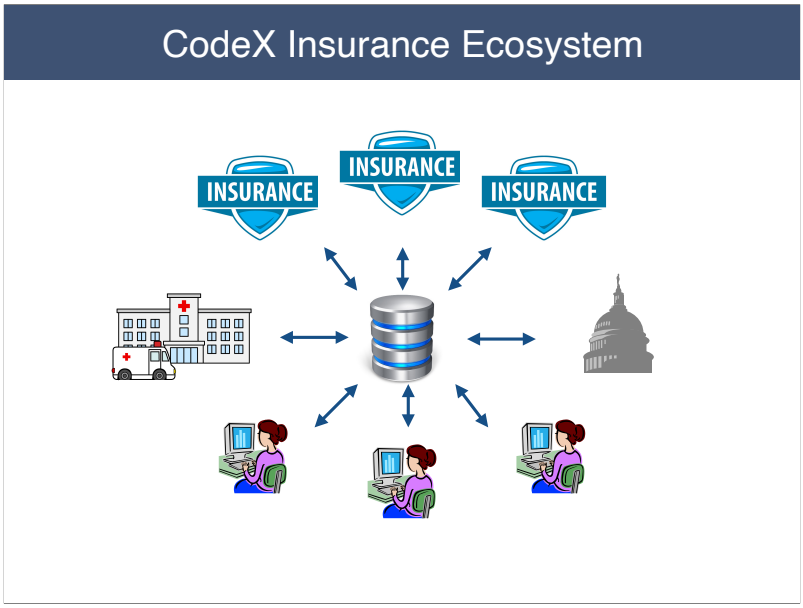
In order to promote the use of this technology, we are developing a public resource called Contractnet. Contractnet is a repository of the datasets, rulesets, and software we are using and making available for use by others. GITHUB is a popular Internet site for programmers to use in sharing open-source code with others. We think of Contractnet as the GITHUB for computable insurance contracts.



Although the current prototype is focussed on individuals, we believe that a similar system can be built to aid businesses in analyzing their insurance needs and opportunities.



And we think that insurance companies can benefit from analyzing their portfolios to understand their risks and business opportunities.



Our long-range goal is to create a digital ecosystem for computable insurance policies that would support and include computable insurance systems like these various flavors of CIA. We believe that an ecosystem of this sort can provide benefit for all parties in the insurance ecosystem. It can benefit customers by analyzing their overall coverage and synthesizing portfolios of insurance products that meets their needs with minimal cost. It can help insurance companies increase transparency, decrease meritless claims, detect up-sell opportunities, and craft customized insurance products. And it can help regulators support and enforce insurance regulations.

There is still more work to be done. However, we would like to move some of this work from the laboratory to the real world. We need to see whether it works in practice; and, if not, we need to understand its weaknesses. Luckily, things are moving forward here. Insurance companies, such as AXA, understand the potential benefits of computable insurance contracts, and they are actively investigating way in which they can utilize this technology in their businesses.

Thank you. Questions?