COMPUTABLE CONTRACTS AND INSURANCE: An Introduction

The CodeX Insurance Initiative Working Group[1]

April 15, 2022

Version 3.1

This discussion brief is the first in a series of short discussion papers that will explore topics related to the use of computable contracts in the insurance industry. They are offered by the CodeX Insurance Initiative, a project of CodeX – The Stanford Center for Legal Informatics, as a resource for education, discussion, research and application development. For more information on the Initiative, see https://law.stanford.edu/projects/codex-insurance-initiative/.

Introduction

This paper provides a conceptual overview, explaining the core concepts of computable contracts and briefly touching on the many areas where these approaches can benefit customers, insurance companies, service vendors, and regulators in the insurance industry. It will also identify two of the principal challenges standing in the way of achieving these benefits.

Contracts, Insurance, and Automation

Contracts are critical components in the insurance business. Some go so far as to say *the contract is the product*, although it is probably more correct to say that the contract is *central to the product*. While contracts are important in many other economic sectors, they generally serve a secondary role, regulating the delivery and payment for the actual products or services offered, such as wine, cell phones, or software consulting services. In contrast, insurance providers sell contracts that make promises to pay if certain damaging events occur. While insurers are increasingly offering additional services, such as loss prevention consultation, the core of the product is still fundamentally a contract.

Because contracts sit so centrally in the insurance business, their automation in a software representation will have profound effects for insurance companies, permitting the development of computable insurance products. These will create the direct efficiencies of being able to have most question related to the execution of the contract quickly and accurately available through a computer interface. Is a given situation covered by the insurance contract? How much of the cost of a specific doctor's visit would be covered under a health insurance contract? What coverage would a business interruption policy provide in the case of a natural calamity such as COVID-19? But the advantages don't stop there. Working outward from the automated contract itself, the benefits of computable insurance products extend to the following:

- facilitating marketing and sales, permitting customization and instant quotes
- making the policy more transparent and understandable for consumers and other users, including through a variety of representation modes

[1] This paper is an effort of the CodeX Insurance Initiative Working Group, with principal authors Oliver Goodenough and Susan Salkind, and with much appreciated knowledge and suggestions from Vinay Chaudhri and Raphael Ancellin.

- creating structured data over the contractual life cycle, multiplying the effectiveness of AI applications and other analytics
- clarifying underwriting calculations
- improving, simplifying, and standardizing policy design and pricing, allowing the creation of new products and accelerating the development timeline
- improving policy servicing and administration, both on a recuring basis and when a claim is made
- supporting the internal oversight and external regulation of the company
- creating a broader marketplace for reinsurance and other inter-company transactions

These are all major cost and performance centers for an insurance company. Once the core product – the insurance contract – has been appropriately automated, the benefits of computing can be increased across most of these tasks as well, improving both quality and efficiency for the entire enterprise. Any concerted, centralized effort to apply computational approaches – including machine learning – to the insurance business needs to rest on a foundation of automated computable contracts.

What are Computable Contracts?

A contract is a memorialized agreement by parties creating mutual obligations that are enforceable by law. A computable contract is one that is specified in sufficient detail and clarity to provide unambiguous answers to questions about compliance of clearly specified circumstances with the terms and conditions of the contract. From a pragmatic point of view, computable contracts can be represented in code and provide the basis for computer systems capable of automated execution and compliance management. When used in this way, we can sometimes speak of automated computable contracts.

Our classic notion of "contracts" describes a document, using the recording power of paper and ink and the coercive power of legal recognition and enforcement, (i) to specify a course of behavior and delivery between two or more otherwise independent actors and (ii) to provide a legally enforceable structure of incentives and penalties that makes the occurrence of that course of behavior and delivery a reliable expectation. While we are used to the word-based formulations of traditional contracting, in an automated computable contract, we can program the definitions of the salient events, and a set of if-then-else rules that specify different consequences based on those events - the heart of all contracts.

It is important to distinguish insurance applications based on automated computable contracts from such widely used terms as "InsurTech," "FinTech" and "LegalTech." LegalTech and InsurTech applications often apply technology to automate only portions of a business process, such as marketing or analytics, typically by sitting on top of a legacy core that still uses a natural language, word-based specification for the contract itself. Document assembly uses computers to help put together natural language terms and rules. Online sales platforms for insurance gather information to fill in the blanks in a word-based contract eventually appearing as a pdf. Natural language processing and machine learning mine these word-based texts to find useful patterns. Advances such as these are helpful, but don't unlock the full power of computing for insurance.

Automated computable contracts, by contrast, envision expressing the rights, duties, and processes defined in a contract directly in machine-executable code. Basic research has demonstrated that legal rules generally take a "computational" form reflecting a well-defined logical structure. Traditional approaches use natural language as the coding medium and lawyer brains as the compiler and processor. By migrating much of the expression, interpretation, and application of the contract terms into software and electronic data, the speed, precision, economy, and scalability of digital computing can be applied to most legal processes – including the contracting practices of the insurance industry. While there has been significant investment in natural language processing that will – eventually – enable it to achieve a better semantic understanding of traditional contracts, the most powerful applications for putting this to work will necessarily involve the direct representation of the understanding in code – an automated computable contract once again.

Computable contracts sit within the wider domains of LegalTech, InsurTech, and FinTech, but they have particular potential, permitting much deeper innovation than applications that automate only ancillary aspects of legacy, natural language-based contracting. In one computable contracting approach, a natural language representation of the contract could co-exist along with the computer program representation of the contract. In a fully automated computable contract, by contrast, all clauses of the contract are expressed as a computer program and are available for query, analysis, verification, and automated execution. In this maximal version, a natural language representation of the contract's terms may not even exist, and its human comprehension may instead be supported through a suitable user interface.

Amazon's computerized purchasing provides a familiar example of the power of automated computable contracts. This remote commerce facility has expanded greatly over the COVID pandemic. Its online presence allows the shopper to see and compare a number of possible purchases, with pricing and variations presented through a carefully developed user interface. Amazon's one-click addition to the user interface makes it even simpler. When available, once a choice is made, and drawing on previously entered information on the consumer, the click activates the order. This action creates what is in effect an automated contract of sale and purchase that permits access to a credit card or other online payment mechanism and then triggers, through integration with Amazon's enterprise management systems, a cascade of instructions for fulfilment and delivery, much of which is also automated.

The attractiveness of these transactions and the efficiencies they offer to both buyer and seller have allowed Amazon to capture the opportunities of the pandemic remote-buying market. In the third quarter of 2020, for instance, Amazon's reported revenue increased 37.4% from 2019 to a record $96.15 billion, with a record at $6.33 billion in net revenues, *an increase of 196.7% over the third quarter of 2019*. Other benefits come from the advanced analytics Amazon can apply to its structured data on ordering, fulfilment, and other operational factors. While automated contracting is hardly the only factor in Amazon's success, this business model has been made possible by the *de facto* computable contract sitting at the center of Amazon's online purchasing system. Imagine trying to implement this system if every order came in via an emailed pdf form in natural language. The automated, computable purchase contract is the backbone on which the other advances can be deployed.

Applications to Insurance

Similar improvements are waiting to be realized in insurance. Returning to the list from earlier in the paper, we can summarize some of them:

> *Facilitating marketing and sales, permitting customization and instant quotes:* The expression of policy elements in code can be designed to be modular, with different coverage targets and

terms able to be assembled by consumers via a user interface to meet their specific needs. The automation can make pricing for these customized insurance bundles an instantaneous calculation provided by the machine to the consumer. While human sales assistance will be desirable for some customers, many will prefer the ease and simplicity of an automated approach, particularly in relatively standardized personal lines of business. If a human agent is involved in the sale, their effectiveness will be enhanced by the availability of such automation as well.

Making the policy more transparent and understandable for the consumer and other users, including through a variety of representation modes: The automated computable contract will allow insurance customers to better understand and engage with their contracts, by facilitating the use of scenario testing, FAQ functions, graphic visualizations, and other explanatory tools. Others interacting with the policy, including sales agents, brokers, and adjusters, will benefit as well.

Creating structured data over the contractual life cycle, multiplying the effectiveness of AI applications and other analytics: Because automated computable contracts can be fully based in software, the data which they consume and emit can be structured, tagged, and captured automatically, without having to go through a messy and often imprecise extraction from natural language documents. This data can then be used in a multitude of ways to understand better the needs of consumers, and the operations of the company and of the broader marketplace. The quality of this data will greatly improve the effectiveness of AI and machine learning approaches to finding useful patterns and insights not accessible to human analysis. In constructing the automated contract system and dealing with the attendant data, care will need to be taken to comply with data privacy rules and to provide robust protection against hacks, attacks, and breaches.

Clarifying underwriting calculations: The underwriting process will also benefit from this kind of analysis. As the factors of risk are clarified through advanced data analytics, actuarial calculations will become more precise. Standardization and simplification of policies can also help to provide better insight into and management of the risks and costs of particular underwriting areas.

Improving, simplifying, and standardizing policy design and pricing, allowing the creation of new products and accelerating the development timeline: Putting these improvements together will allow companies to redesign the products they offer, based on consumer preference, clearer actuarial modeling, and advanced analytics. These improvements can be fed back into the mix, and be tracked, in their turn, for customer acceptance and profitability. An automated computable-contract spine will also facilitate the creation of new products, such as micro insurance, imbedded insurance, IoT coverage, and autonomous and continuous variable underwriting. This customization can go hand in hand with a simplification and standardization of the policy elements themselves, which will, in turn, improve underwriting and realize operational savings.

Improving policy servicing and administration, both for the overall operation, and for handling the individual claim: Policy administration generally, and claims settlement in particular, are significant cost centers for most insurers. A computable contract approach can permit the company to automate more of these tasks and greatly improve the performance of humans working on the remainder.

Supporting the internal oversight and external regulation of the company: Having the policies of the company expressed in automated computable contracts will greatly assist management's oversight on its portfolio of risks. Future exposure can be explored by directly stress testing the company's policies against a variety of event combinations. Take, for instance, the COVID pandemic. It has revealed surprising exposures for many insurers, correlated across countries and business sectors in ways not always anticipated. If the obligations of insurers facing these exposures had been modeled in advance by testing their business interruption and related policies against simulated outbreaks, better planning could have occurred. Regulatory reporting will also be improved and made easier if the data produced by automated computable contracts is properly gathered, curated, and reported.

Creating a broader marketplace for reinsurance and other inter-company transactions: So far, this discussion has focused on advantages within a particular enterprise. There are also significant potential benefits across the marketplace as a whole, particularly if interoperable approaches to data representation and software are shared broadly. For instance, this could allow efficiency and quality improvements in reinsurance transactions and in other means of pooling and sharing risk.

Implementation Challenge 1: Technological Approaches

The technological requirements for creating a computable contract platform for the insurance enterprise are challenging, but specifiable. At a basic level, we require a programming language for contracts, a well-developed approach to data representation, a library of local logic patterns that could be expressed in that language using the data, the interfaces through which users could author and consume computable contracts, and the algorithmic tools for different tasks of interest. We consider each of these tasks in greater detail.

The computable contracts research community has been pursuing a variety of programming approaches to automation, including functional, imperative, and logic programming. Each of these approaches has merits and contributes to realizing the vision of computable contracts. CodeX has advocated the use of logic programming for the following reasons:

(a) Natural language representations of traditional contracts naturally express clauses as rules which more directly map to the statements in a logic program.

(b) A logic program naturally separates the *what* from *how*, and thus the statements need not be stated in a fixed order, but rather can be applied in any order required by novel situations.

(c) Logic programs lend themselves more naturally to multiple computational tasks including question-answering, formal verification, and automated execution.

The obligations common to a particular type of business generally have a particular structure, and, in practice, such a structure often repeats across a whole family of contracts for that business. We refer to such a repeating structure of a set of contracts as its "local logic." Local logic is sometimes reusable across domains. For example, nearly every loan agreement has a loan amount, a time duration, a rate of interest, a lender, and a borrower. A loan agreement generally goes through stages in which it is first activated, money is lent, interest accrues, and then the loan is paid off. In some cases, payments may be late, and the contract may go into default. Analogously, insurance contracts generally include coverages and amounts, a set of payment triggers, a time duration, conditions, deductibles and exclusions, and procedures for the filing and administration of claims. The contract also goes through the stages of activation, and claims filing and payment. In some cases, the policy may be canceled because some condition of the insurance contract is violated. In order to avoid repetitive programming, large clause libraries expressed in computer code can be created based on such local logic of contracts that are tailored to particular facts and risks. Such clause libraries allow a modular approach, with reusable elements, to inform the programming of computable contracts, creating efficiencies in the assembly of customized contracts. Existing policy and clause categories used in the wording of natural language contracts already reflect this kind of categorization. Extending it to software-based elements is not that big a reach.

Also critical are user interfaces for contract construction in both B2B and B2C environments. Recall that in the minimally automated computable contract, natural language and the computer program representation of a contract co-exist. There needs to be a suitable contract authoring interface through which the computer program representation can be authored and correlated to various natural language versions, elements, and representations. The authoring interface must reflect how various parties in the value chain understand the needs and promises of the insurance bargain. The data schema embedded in the application should also be developed with the practices of insurance in mind and with an eye to interfacing with the business management systems that will carry the contract's instructions for implementation throughout the enterprise. In the maximally automated computable contract, all of the contract clauses exist as computer code. Therefore, the authoring interface should present it in multiple forms or representations for consumption by each key party with whom it must interact. For example, while a lawyer may wish to see a line-by-line natural language conversion of the computer code, a customer may prefer a flow chart-like view that highlights various contract parameters.

Finally, task-specific reasoning algorithms must be developed to perform computations over the contract. Such an algorithm could develop new designs of an insurance policy. For example, motivated by the COVID pandemic, an insurance company might wish to revise its portfolio of contracts to add special new provisions. A contract analysis algorithm can help perform what-if analyses for scenarios such as the impact of different types and levels of coverage and deductibles.

Real examples exist of each of these approaches. While most still have limited application and penetration, they are advanced enough to demonstrate the feasibility of applying computable contracts to each of the uses discussed.

Implementation Challenge 2: Fostering Adoption within a Firm

Just because a business advance is possible and would bring benefits does not mean that it will be adopted. There is much empirical and anecdotal evidence describing hurdles in the way of beneficial, if disruptive, innovation. The insurance business has been quite traditional in many of its practices. It can be a challenge to persuade leaders and other key stakeholder groups that they should embrace a new way of creating, marketing, and administering the company's core product. That said, the impact of technology on other industries suggests a lesson: transform, or face declining market share and profitability.

This resistance to change is not just a matter of conservatism. Updating the IT legacy system of an insurance company is a difficult, multimillion dollar, and multiyear project per business unit. Many companies have grown through mergers and acquisitions, and the acquired units often bring with them disparate, aging systems, often inherited from previous M&A activities. Implementing an automated contracting system can require fixing this patchwork to get full benefits. Because of these costs, it is not uncommon to see companies waiting until the very last minute before adopting change, at worst because the systems are so dated that no one understands how to maintain them. On the other hand, the potential returns from a conversion to computable contracts, even if hard to fully quantify, could help to persuade companies to undertake such a conversion.

As is often the case, a productive approach to managing this change may involve finding particular business sectors and territories where pilots can be run, advancing the learning curve for the enterprise without putting the broader business at risk. Areas of business where the pain points hurt most or where the benefits can be most quickly realized are good early targets. The generally simpler contracts of personal lines make this an attractive business area for automation, although more complex commercial policies pose addressable challenges. Lessons learned and techniques developed in these pilots can then be broadened out into the company and the industry more generally. The process will take a bit of time, but failing to start on it can put an insurer in a situation to catch up or fail.

Opposition may also come from other ecosystem stakeholders, such as agents and brokers. They too, however, face the dilemma of modernizing or becoming irrelevant. Keeping them focused on this can be helpful.

Implementation Challenge 3: Operational Security and Legal and Regulatory Compliance

Moving to a more highly automated system for contracting and for collecting and using data will raise questions of compliance and regulation that will need to be addressed. Insurance is a highly regulated business, and, in many instances, policy and other contracts must receive regulatory approval. The filing and review systems in place for approving natural language policies and the contracts that implement them will need to expand to deal with code-based versions. Engaging regulators in the process of developing automated computable-contract innovation may be helpful here as in other areas of digital transformation. Because of the potential benefits to customers, insurers, and the industry as a whole, the end result should be positive, but the process may take time.

In addition to other regulatory concerns, a company's automation of computable contracts will, of course, also need to meet requirements for data privacy and security, and for operational security as a whole. As with other innovation efforts, these concerns need to be addressed from the very beginning, incorporating privacy and security by design approaches that meet legal and security requirements while still allowing productive data analytics and direct customer access.

Implementation Challenge 4: Creating a Community and Ecosystem

In order to enable the development of industry-wide interoperability and to share knowledge and best practices, both the insurance industry as a whole and individual entities within it will benefit from the establishment of community efforts around computational contract approaches. The CodeX Insurance

Initiative is intended, in part, to provide an academic home for this discussion, knowledge-sharing, and ecosystem-building.

Conclusions

The adoption of automated computable contracts by the insurance industry will create significant improvements in efficiency, and in customer experience and outcomes. It can also lead to significant innovations in data collection and analysis, policy design and administration, actuarial accuracy, oversight and regulation, and shared marketplaces for risk. Looking beyond a particular company, interoperability more widely across the industry will make many benefits possible, including better secondary markets for pooling and sharing risk.

There are potential obstacles to capturing these benefits. The technology is available, but implementation needs to be capably handled. The conservative nature of the insurance industry may slow adoption of beneficial innovation. And regulatory and data and security concerns need to be met. Developing communities for the permissible sharing of information, solutions, and open-source technology can help to speed the development process and minimize risk along the way.

Future papers in this series will explore a number of these opportunities and concerns, together with the computer science methods and standards that will enable the implementation of insurance products and systems based on automated computable contracts.