

Synthesising Implicit Contracts

Gordon J. Pace¹ and Fernando Schapachnik²

¹ University of Malta

² Universidad de Buenos Aires, Argentina

Abstract. In regulated interactive systems, one party's behaviour may impose restrictions on how others may behave when interacting with it. These restrictions may be seen as *implicit contracts* which the affected party has to conform to and may thus be considered inappropriate or excessive if they overregulate one of the parties. In [5], we have characterised such implicit contracts and present an algorithmic way of synthesising them using a formalism based on contract automata to regulate interactive action-based systems. In this presentation, we outline the problem and future extensions of the work we are currently exploring.

1 Introduction

Consider the contract that binds a customer and a bank, which stipulates that opening new accounts is free of charge. And yet, at the moment of opening an account, the bank requires the release of personal information and allowance to send the customer promotional material. The bank is not strictly breaching the contract, but maybe it is asking “too much”. Can this “too much” be quantified? As another example consider an airline, which compensates for missed connections due to delays by providing the traveller with food and lodging. However, the airline has a policy of not providing this service unless the customers explicitly demands for it. In a way, the airline is turning its unconditional obligation of providing aid into a conditional or restricted one: given that the customer asks for help, support will be provided.

In interactive systems involving different parties, the behaviour of each party is inherently affected by the behaviour of the others. In particular, other parties may restrict or enforce behaviour resulting in the affected party having to behave in a different manner than if it were to act independently. In interactive systems regulated by contracts, each party thus may be seen to be restricted through the explicitly agreed upon contracts, but furthermore by the implicit constraints induced through the nature of interaction and the other parties' behaviour. These implicit constraints can be seen as ‘invisible’ (or unspoken) agreements or contracts a party has to agree to and adhere to if they choose to participate in the interaction.

As the imposed behaviour gets stricter, it drifts from being compliant up to the point of being close to breaching the contract. Can this drifting be measured? As a first step, in this paper we develop techniques so that, given two interacting parties bounded by a contract, we can infer the implicit contract being enforced by one party on the other.

This approach is useful also during contract negotiation: given multiple service providers, a party may base her choice of provider not only upon the signed agreement, but also taking into consideration the implicit contract imposed on her.

2 Results

In [5], we have formalised the notion of *implicitly enforced contract* — corresponding to a contract which one party has to satisfy due to its interaction with the other party. For instance, if the *bank* party requires synchronisation over an action *register* before allowing the *client* party to engage in an action *transfer*, the implicitly enforced contract would include an obligation on the *client* to perform *register* before proceeding.

To formalise these notions, we have used *contract automata* [6], a formalism based on the notion of synchronising automata, that provides well understood and clear semantics to model interaction among parties. An automata-based formalism allows to model not only deontic formulae but also the choices that each party has and how decisions affect the other parties.

Contract automata give us a natural definition for a partial order of contract strictness. Using this notion, we have presented an algorithm to calculate the maximal implicitly enforced contract which can be proved to be unique.

3 Future Directions

Although contracts are a long-covered topic in deontic literature (e.g., [2–4]), we are not aware of work on synthesising implicit contracts. Process calculi (e.g., [1]) or even petri nets (e.g., [7]) have been used before to model “contracts”. However, the term contract is used there in the sense of *interface*, as way to guarantee services interoperability, but they are not rich enough for other types of contracts because they do not support deontic operators.

Although the concept of implicit contracts is interesting in itself, it becomes more interesting when one starts to compare this implicitly enforced behaviour to what a given contract demands, as a tool to highlight not only potential breaches but also subtle differences. Such a comparison permits the view of a continuum going from plain incompatibility to full compliance, with a lattice of possibilities in between — for instance, when one party restricts a permission of the other party by only allowing the permitted action after certain extra actions are performed.

To further explore implicitly enforced behaviour, synthesis for non-deterministic systems, is still an open problem. Future research directions also includes the dual of implicitly enforced behaviours, *intrinsic well-behaviour*: given the behaviour of a system, induce a contract which it obeys under all circumstances. Such a notion could, for instance, be useful to use to figure out how much more one can ask from the other party without them having to change their behaviour.

References

1. M. Bravetti and G. Zavattaro. Contract based multi-party service composition. In *International Symposium on Fundamentals of Software Engineering*, pages 207–222. Springer, 2007.
2. G. Governatori and Z. Milosevic. Dealing with contract violations: formalism and domain specific language. In *EDOC Enterprise Computing Conference, 2005 Ninth IEEE International*, pages 46–57. IEEE, 2005.

3. O. Marjanovic and Z. Milosevic. Towards formal modeling of e-contracts. In *Proceedings of the 5th IEEE International Conference on Enterprise Distributed Object Computing, EDOC '01*, pages 59–, Washington, DC, USA, 2001. IEEE Computer Society.
4. G. Pace and F. Schapachnik. Contracts for interacting two-party systems. In *FLACOS 2012: Sixth Workshop on Formal Languages and Analysis of Contract-Oriented Software*, sep 2012.
5. G. Pace and F. Schneider. Synthesising implicit contracts. In *Proceedings of the XIV International Conference on Artificial Intelligence and Law (ICAIL 2013)*. ACM, 2013.
6. G. J. Pace and F. Schapachnik. Permissions in contracts, a logical insight. In *JURIX*, pages 140–144, 2011.
7. W. M. Van Der Aalst, N. Lohmann, P. Massuthe, C. Stahl, and K. Wolf. From public views to private views—correctness-by-design for services. In *Web Services and Formal Methods*, pages 139–153. Springer, 2008.