Two notions of sub-behaviour for session-based client/server systems: 10 Years Later

Franco Barbanera
barba@dmi.unict.it
Dipartimento di Matematica e Informatica
University of Catania, Italy

Ugo de'Liguoro
ugo.deliguoro@unito.it
Dipartimento di Informatica
University of Torino, Italy

CCS CONCEPTS

• **Theory of computation** → *Process calculi*; *Type theory*; *Operational semantics.*

KEYWORDS

Behavioural Type, Session Contract, Session Type, Subcontract

ACM Reference Format:
Franco Barbanera and Ugo de'Liguoro. 2020. Two notions of sub-behaviour for session-based client/server systems: 10 Years Later. In *22nd International Symposium on Principles and Practice of Declarative Programming (PPDP '20), September 8–10, 2020, Bologna, Italy.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3414080.3414082

We wish to begin this short overview of our work on relating session types to contract theory by expressing our gratitude to the PPDP Steering Committee, for awarding our paper [1] with the "PPDP 10 Year Most Influential Paper Award" of this year. At the same time we would like to state that we are not expecting to have introduced new ideas, nor to have opened unforeseen perspectives on the subject. Rather we have contributed to frame ideas coming from different sources in a couple of conceptually simple constructions, participating to a fruitful debate and a choral work lasting more than twenty years, and still going on.

Context. The issue of specifying program behaviour in a modular and flexible way, in order to adapt and reuse code in different contexts, is of paramount importance in the field of software development. Among specification techniques, types have been devised as a good compromise between formalizing syntactical constraints that can be efficiently checked at compile time, and conveying abstract mathematical ideas into the design of complex software systems.

Introduced in order to improve flexibility of type specifications, the idea of subtyping dates back to the 1960s. Roughly, it reminds (but doesn't coincide with) the notion of subset, where values of some type A can be seen as values of type B whenever $A <: B$, namely A is a subtype of B. Such a view, stemming from functional programming, has been successfully adapted to other paradigms, such as procedural and object-oriented programming, based on the idea of substitutivity of code representing values of some subtype into contexts expecting code computing values of a supertype of its.

Moving to communication centered systems and concurrent programming languages, the notion of value is not central anymore. Rather, what matters is the structure of the interaction, which is often called the *behaviour* of a system. Behaviours are far more abstract and rather elusive entities, however; natural models of behaviours are *labelled transition systems* (LTS), where they are interpreted as equivalence classes under various kinds of relations like bisimulation, to abstract away from inessential differences among LTS. In contrast to their mathematical elegance, bisimulation and related notions are often too strong and undecidable, whereas coarser principles are more amenable to formal systems at the basis of efficient tools for static program analysis.

In this context, two formalisms have recently arisen: *session types*, introduced in the seminal work [33], and *contract theories* [27, 30]; see [34] for a survey and a comparison of these approaches. The former is a type system, where process-like types are assigned to *sessions* (represented by private channels), abstractly representing communication protocols. Contract theories are, instead, essentially subcalculi of Milner's CCS without τ [39], formalizing the input-output behaviour of autonomous concurrent entities, among which a notion of *compliance* is defined: see [17] and [37] for a survey and a comparison of several (non equivalent) definitions of this concept. A client satisfying a contract ρ (henceforth identified with ρ) complies with a server respecting σ (and identified with σ), written $\rho \dashv \sigma$, if any action from ρ is "matched" by a dual action by σ, so that any possible interaction among ρ and σ will never prevent ρ from completing. Clearly different definitions of matching induce different relations of compliance.

With respect to the concept of *strong-compliance* in [24], consisting of deadlock and livelock freeness of the composition of two processes, compliance in the above sense is a weaker notion; however it has a rich mathematical theory and enough expressive power to discriminate well-behaved compositions of processes. As shown in [35, 40], one can formulate a natural notion of sub-contract relation inspired to must-preorder from *testing semantics* of CCS [32, 38] that is based on the concept of observation via experiments, although the two are subtly distinct relations [18].

Following [43], Gay and Hole proposed in [31] a theory of subtyping for session types, that was explicitly intended as a formalization of the substitutivity principle, but still lacking of a clear semantical intuition. Investigating a semantical understanding of the notion of sub-behaviour in a context of contracts was then an appealing idea to provide a mathematical model of sub-typing for process calculi in the spirit of [26], while retaining a simple formal system to reason about types and processes.

PPDP '20, September 8–10, 2020, Bologna, Italy
© 2020 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-8821-4/20/09.
https://doi.org/10.1145/3414080.3414082

Contribution. The PPDP'10 paper "Two notions of sub-behaviour for session-based client/server systems" [1] was conceived at the confluence of the lines of thought sketched above. First we selected a proper sublanguage of contracts perfectly mirroring session types, which otherwise do not correspond to each other [36]. We called the expressions of this language *session behaviors* (later called *session contracts* e.g. in [22]) and equipped them with an LTS semantics. We also defined the relation $\rho \dashv \sigma$ using a relaxed requirement about termination of ρ, which can now engage an unbounded interaction with σ, but maintained the bias toward the client from [35], that can freely abandon the interaction in contrast to the server. This yields two natural preorders among contracts, depending on the point of view of a successful interaction by a server and a client respectively: the *server sub-behaviour* relation, according to which we said that the server σ is less permissive than σ', written $\sigma \preceq_s \sigma'$ if any client ρ that is compliant with σ is such with σ'; symmetrically we defined the *client sub-behaviour* relation, according to which a client ρ is at least as demanding as ρ', written $\rho \preceq_c \rho'$ if all the servers with which ρ is compliant are such that ρ' is compliant with them. As a consequence we proved that the intersection $\preceq_s \cap \preceq_c$ is a model of Gay and Hole subtyping for session types without delegation (see below).

Extending the theory of the sub-behaviour relations to higher-order contracts, that are able to exchange contracts as values in a communication called *delegation*, revealed to be much more challenging than expected. Indeed the proof sketched in [1] had a bug, so that in [3] we resorted to a stratification technique and a careful analysis of higher-order behavioural contracts, revealing that to obtain a model of the full subtyping system in [31] one has to treat exchanged contracts in a delegation as invariant w.r.t. the sub-behaviour relation. We called this third notion the *peer sub-behaviour* relation, and *CSP-subtyping* the subtyping system for session types for which it is a sound and complete model, that properly extends Gay and Hole's theory.

Developments and impact. It is worth observing that the quest for a semantic interpretation of session subtyping was also the main motivation of the previous PPDP'19 10-Year-award paper [25]. Indeed many contributions on the subject have appeared since then. The client and peer preorders we began to study were subsequently investigated in [18, 19] for must testing. A fully-abstract characterisation of client preorders due to must and compliance testing was then developed and analysed in [21]. The theory of higher-order contracts has been studied in depth in [20], including similar results to ours, but without using the complex machinery of stratification and adopting a richer language. Even a timed version of first-order fragment of behavioural contracts has been considered in the context of a more intentional analysis of protocols in [14].

As shown in [17, 34, 37], in the subsequent years the investigation about well-behaved composition of processes has focused on alternative views of the notion of compliance and their induced subtyping relations, both to improve flexibility in composing two processes and because of the growing interest about multi-party communication, both synchronous and asynchronous.

An example is the notion of orchestrated compliance in [40, 41], based on the assumption that client/server interactions may be mediated by *orchestrators*. The aim of an orchestrator is guaranteeing "client satisfaction" by affecting - by means of (finite) buffering capabilities - the way clients and servers might communicate to each other. In [2, 13] we investigated further into the notion of orchestrated compliance and subcontract by considering orchestrators with unbounded buffers and interaction-bypassing capabilities.

Another route towards the goal of "client satisfaction" is to change the semantics of contracts so that they can adapt to each other by means of a rollback mechanism. These are the *retractable contracts* that Dezani, Lanese and ourselves have proposed and investigated in [8, 9] together with the notion of compliance they induce. Another possible semantics of contracts is the speculative one, where in a choice all branches can be executed in parallel. We investigated the induced notion of compliance (and the corresponding subcontract relation) in [11, 12] and proved speculative and retractable compliance relations to be equivalent.

Since both backtracking and orchestration are means to affect the interactions between two contracts, we defined in [4] a notion of compliance subsuming both retractable and orchestrated compliance: *affectible compliance.* Affectible compliance, however, is not a new relation, rather it coincides with both the former, but seen as different instances of the same concepts. As a matter of fact the proof, inspired by [15, 16], consists of a game-theoretic interpretation of compliance, that we think to be of interest on its own.

As further evidence of the fruitful interaction and cross contamination between the theories of contracts and of session types, we adapted the idea of retractable contracts and orchestrated compliance to session types in [5]. We provided a calculus where possible stuck states during an interaction on a channel can be avoided by means of backtracking or, equivalently, by an orchestrating process. Moreover, the latter can be synthesised out of the session type associated to the channel.

We conclude this fatally incomplete excursus by mentioning two directions of research on related subjects which the authors have recently contributed to. First let us mention the automata-theoretic modeling of interaction among concurrent programs of *Communicating Finite State Machines*, CFSM, introduced in [23]. To our knowledge, the first contribution relating CFSM and session types is [29], focused on the study on how, and within which limits, properties of deadlock and error freedom that had be proven to be computationally hard by means of the CFSM model, can be expressed by a decidable type system like session types. As usual, the trick to prevent undecidability is to guarantee a priori the desired properties to hold of processes respecting suitable type constraints. It cannot be then by mere chance that the restrictions we imposed on session contracts are among those required on "compatible" CFSM systems, which are sufficient conditions to safely compose communicating automata according to the approach that Hennicker and ourselves devised in [6]. Together with Dezani, Lanese and Tuosto, the first author applied this approach also to multiparty session types [7, 10], where the same restrictions implicitly hold and the notion of compatibility is relaxed, resembling that of equivalence induced by our peer-subcontract relation.

A second line of work deals directly with the theory of subtyping. Stemming from Padovani's work on fair testing and subtyping of multiparty session types [42], the system of *mail boxes types* in [28] introduces a type system for reasoning on protocol conformance and deadlock freedom in networks of processes that communicate through unordered mailboxes. An interesting novelty of the approach is the use of commutative Kleene algebras as the underlying mathematical model to "loosen" too strict syntactical constraints normally found both in session types and contract theories.

Lastly, a thought of gratitude to Mariangiola Dezani for her immutable friendly support.

REFERENCES

[1] Franco Barbanera and Ugo de'Liguoro. 2010. Two notions of sub-behaviour for session-based client/server systems. In *Proceedings of the 12th International ACM SIGPLAN Conference on Principles and Practice of Declarative Programming, July 26-28, 2010, Hagenberg, Austria*, Temur Kutsia, Wolfgang Schreiner, and Maribel Fernández (Eds.). ACM, 155–164. https://doi.org/10.1145/1836089.1836109

[2] Franco Barbanera and Ugo de'Liguoro. 2014. Loosening the notions of compliance and sub-behaviour in client/server systems. In *Proceedings ICE 2014 (EPTCS)*, Ivan Lanese, Alberto Lluch-Lafuente, Ana Sokolova, and Hugo Torres Vieira (Eds.), Vol. 166. 94–110. https://doi.org/10.4204/EPTCS.166.10

[3] F. Barbanera and U. de'Liguoro. 2015. Sub-behaviour relations for session-based client/server systems. *MSCS* 25, 6 (2015), 1339–1381.

[4] Franco Barbanera and Ugo de'Liguoro. 2017. Retractability, games and orchestrators for session contracts. *Log. Methods Comput. Sci.* 13, 3 (2017). https://doi.org/10.23638/LMCS-13(3:15)2017

[5] Franco Barbanera and Ugo de'Liguoro. 2017. Session Types for Orchestrated Interactions. In *Proceedings ICE 2017 (EPTCS)*, M.Bartoletti, L.Bocchi, L.Henrio, and S.Knight (Eds.), Vol. 261. 17–36. https://doi.org/10.4204/EPTCS.261.5

[6] Franco Barbanera, Ugo de'Liguoro, and Rolf Hennicker. 2019. Connecting open systems of communicating finite state machines. *J. Log. Algebraic Methods Program.* 109 (2019). https://doi.org/10.1016/j.jlamp.2019.07.004

[7] Franco Barbanera and Mariangiola Dezani-Ciancaglini. 2019. Open Multiparty Sessions. In *Proceedings ICE 2019 (EPTCS)*, M.Bartoletti, L.Henrio, A.Mavridou, and A.Scalas (Eds.), Vol. 304. 77–96. https://doi.org/10.4204/EPTCS.304.6

[8] Franco Barbanera, Mariangiola Dezani-Ciancaglini, and Ugo de'Liguoro. 2016. Reversible client/server interactions. *Formal Asp. Comput.* 28, 4 (2016), 697–722. https://doi.org/10.1007/s00165-016-0358-2

[9] F. Barbanera, M. Dezani-Ciancaglini, I. Lanese, and U. de'Liguoro. 2016. Retractable Contracts. In *PLACES 2015 (EPTCS)*, Vol. 203. Open Publishing Association, 61–72.

[10] Franco Barbanera, Mariangiola Dezani-Ciancaglini, Ivan Lanese, and Emilio Tuosto. 2020, to appear. Composition and Decomposition of Multiparty Sessions. *Journal of Logical and Algebraic Methods in Programming* (2020, to appear).

[11] Franco Barbanera, Ivan Lanese, and Ugo de'Liguoro. 2017. Retractable and Speculative Contracts. In *Coordination Models and Languages - 19th IFIP WG 6.1 International Conference, COORDINATION 2017, Proceedings (LNCS)*, Jean-Marie Jacquet and Mieke Massink (Eds.), Vol. 10319. Springer, 119–137. https://doi.org/10.1007/978-3-319-59746-1_7

[12] Franco Barbanera, Ivan Lanese, and Ugo de'Liguoro. 2018. A theory of retractable and speculative contracts. *Sci. Comput. Program.* 167 (2018), 25–50. https://doi.org/10.1016/j.scico.2018.06.005

[13] Franco Barbanera, Steffen van Bakel, and Ugo de'Liguoro. 2017. Orchestrated session compliance. *J. Log. Algebr. Meth. Program.* 86, 1 (2017), 30–76. https://doi.org/10.1016/j.jlamp.2016.08.002

[14] Massimo Bartoletti, Tiziana Cimoli, and Maurizio Murgia. 2017. Timed Session Types. *Log. Methods Comput. Sci.* 13, 4 (2017). https://doi.org/10.23638/LMCS-13(4:25)2017

[15] Massimo Bartoletti, Tiziana Cimoli, and G. Michele Pinna. 2014. A note on two notions of compliance. In *Proceedings ICE 2014, (EPTCS)*, Ivan Lanese, Alberto Lluch-Lafuente, Ana Sokolova, and Hugo Torres Vieira (Eds.), Vol. 166. 86–93. https://doi.org/10.4204/EPTCS.166.9

[16] Massimo Bartoletti, Tiziana Cimoli, G. Michele Pinna, and Roberto Zunino. 2016. Contracts as games on event structures. *J. Log. Algebraic Methods Program.* 85, 3 (2016), 399–424. https://doi.org/10.1016/j.jlamp.2015.05.001

[17] Massimo Bartoletti, Tiziana Cimoli, and Roberto Zunino. 2015. Compliance in Behavioural Contracts: A Brief Survey. In *Programming Languages with Applications to Biology and Security - Essays Dedicated to Pierpaolo Degano on the Occasion of His 65th Birthday (LNCS)*, Chiara Bodei, Gian Luigi Ferrari, and Corrado Priami (Eds.), Vol. 9465. Springer, 103–121. https://doi.org/10.1007/978-3-319-25527-9_9

[18] Giovanni Bernardi and Matthew Hennessy. 2013. Compliance and Testing Preorders Differ. In *SEFM 2013 Collocated Workshops: BEAT2, WS-FMDS, FM-RAIL-Bok, MoKMaSD, and OpenCert, Revised Selected Papers (LNCS)*, S.Counsell and M.Núñez (Eds.), Vol. 8368. Springer, 69–81. https://doi.org/10.1007/978-3-319-05032-4_6

[19] Giovanni Bernardi and Matthew Hennessy. 2015. Mutually Testing Processes. *Log. Methods Comput. Sci.* 11, 2 (2015). https://doi.org/10.2168/LMCS-11(2:1)2015

[20] Giovanni Bernardi and Matthew Hennessy. 2016. Using higher-order contracts to model session types. *Logical Methods in Computer Science* 12, 2 (2016). https://doi.org/10.2168/LMCS-12(2:10)2016

[21] Giovanni Tito Bernardi and Adrian Francalanza. 2018. Full-abstraction for client testing preorders. *Sci. Comput. Program.* 168 (2018), 94–117. https://doi.org/10.1016/j.scico.2018.08.004

[22] G. T. Bernardi and M. Hennessy. 2016. Modelling session types using contracts. *Mathematical Structures in Computer Science* 26, 3 (2016), 510–560.

[23] Daniel Brand and Pitro Zafiropulo. 1983. On Communicating Finite-State Machines. *J. ACM* 30, 2 (1983), 323–342. https://doi.org/10.1145/322374.322380

[24] Mario Bravetti and Gianluigi Zavattaro. 2007. A Theory for Strong Service Compliance. In *Coordination Models and Languages, 9th International Conference, COORDINATION 2007, Proceedings (Lecture Notes in Computer Science)*, Amy L. Murphy and Jan Vitek (Eds.), Vol. 4467. Springer, 96–112. https://doi.org/10.1007/978-3-540-72794-1_6

[25] Giuseppe Castagna, Mariangiola Dezani-Ciancaglini, Elena Giachino, and Luca Padovani. 2019. Foundations of Session Types: 10 Years Later. In *Proceedings of PPDP 2019*, Ekaterina Komendantskaya (Ed.). ACM, 1:1–1:3. https://doi.org/10.1145/3354166.3356340

[26] Giuseppe Castagna and Alain Frisch. 2005. A Gentle Introduction to Semantic Subtyping. In *Automata, Languages and Programming ICALP 2005, Proceedings (LNCS)*, Luís Caires, Giuseppe F. Italiano, Luís Monteiro, Catuscia Palamidessi, and Moti Yung (Eds.), Vol. 3580. Springer, 30–34.

[27] Giuseppe Castagna, Nils Gesbert, and Luca Padovani. 2009. A theory of contracts for Web services. *ACM Trans. Program. Lang. Syst.* 31, 5 (2009), 19:1–19:61. https://doi.org/10.1145/1538917.1538920

[28] Ugo de'Liguoro and Luca Padovani. 2018. Mailbox Types for Unordered Interactions. In *32nd European Conference on Object-Oriented Programming ECOOP 2018, Proceeding (LIPIcs)*, Todd D. Millstein (Ed.), Vol. 109. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 15:1–15:28. https://doi.org/10.4230/LIPIcs.ECOOP.2018.15

[29] Pierre-Malo Deniélou and Nobuko Yoshida. 2013. Multiparty Compatibility in Communicating Automata: Characterisation and Synthesis of Global Session Types. In *Automata, Languages, and Programming ICALP 2013, Proceedings, Part II (LNCS)*, F.V. Fomin, R.Freivalds, M.Z. Kwiatkowska, and D.Peleg (Eds.), Vol. 7966. Springer, 174–186. https://doi.org/10.1007/978-3-642-39212-2_18

[30] Cédric Fournet, C. A. R. Hoare, Sriram K. Rajamani, and Jakob Rehof. 2004. Stuck-Free Conformance. In *Computer Aided Verification CAV 2004, Proceedings (LNCS)*, Rajeev Alur and Doron A. Peled (Eds.), Vol. 3114. Springer, 242–254. https://doi.org/10.1007/978-3-540-27813-9_19

[31] Simon Gay and Malcolm Hole. 2005. Subtyping for Session Types in the Pi-Calculus. *Acta Informatica* 42, 2/3 (2005), 191–225.

[32] Matthew Hennessy. 1988. *Algebraic theory of processes.* MIT Press.

[33] K. Honda, V. T. Vasconcelos, and M. Kubo. 1998. Language Primitives and Type Disciplines for Structured Communication-based Programming. In *ESOP (LNCS)*, Vol. 1381. Springer, 22–138.

[34] Hans Hüttel, Ivan Lanese, Vasco T. Vasconcelos, Luís Caires, Marco Carbone, Pierre-Malo Deniélou, Dimitris Mostrous, Luca Padovani, António Ravara, Emilio Tuosto, Hugo Torres Vieira, and Gianluigi Zavattaro. 2016. Foundations of Session Types and Behavioural Contracts. *ACM Comput. Surv.* 49, 1 (2016), 3:1–3:36. https://doi.org/10.1145/2873052

[35] Cosimo Laneve and Luca Padovani. 2007. The *Must* Preorder Revisited. In *CONCUR 2007 - Concurrency Theory, 18th International Conference, CONCUR 2007, Lisbon, Portugal, September 3-8, 2007, Proceedings (Lecture Notes in Computer Science)*, Luís Caires and Vasco Thudichum Vasconcelos (Eds.), Vol. 4703. Springer, 212–225. https://doi.org/10.1007/978-3-540-74407-8_15

[36] Cosimo Laneve and Luca Padovani. 2008. The Pairing of Contracts and Session Types. In *Concurrency, Graphs and Models, Essays Dedicated to Ugo Montanari on the Occasion of His 65th Birthday (LNCS)*, Pierpaolo Degano, Rocco De Nicola, and José Meseguer (Eds.), Vol. 5065. Springer, 681–700. https://doi.org/10.1007/978-3-540-68679-8_42

[37] Maurizio Murgia. 2019. A Note On Compliance Relations And Fixed Points. In *Proceedings ICE 2019, (EPTCS)*, Massimo Bartoletti, Ludovic Henrio, Anastasia Mavridou, and Alceste Scalas (Eds.), Vol. 304. 38–47. https://doi.org/10.4204/EPTCS.304.3

[38] Rocco De Nicola and Matthew Hennessy. 1983. Testing Equivalence for Processes. In *ICALP (LNCS)*, Vol. 154. Springer, 548–560.

[39] Rocco De Nicola and Matthew Hennessy. 1987. CCS without tau's. In *TAPSOFT, Vol.1 (LNCS)*, Vol. 249. Springer.

[40] Luca Padovani. 2009. Contract-Based Discovery and Adaptation of Web Services. In *SFM (LNCS)*, Vol. 5569. Springer, 213–260.

[41] L. Padovani. 2010. Contract-Based Discovery of Web Services Modulo Simple Orchestrators. *Theoretical Computer Science* 411 (2010), 3328–3347.

[42] Luca Padovani. 2016. Fair subtyping for multi-party session types. *Math. Struct. Comput. Sci.* 26, 3 (2016), 424–464. https://doi.org/10.1017/S096012951400022X

[43] Benjamin C. Pierce and Davide Sangiorgi. 1993. Typing and Subtyping for Mobile Processes. In *Logic in Computer Science.* Full version in *Mathematical Structures in Computer Science* , Vol. 6, No. 5, 1996.