Time as a First-Class Citizen
The ΔQ Systems Design Project

Kevin HAMMOND[1], Peter W. THOMPSON[1,2], Neil J. DAVIES[1,2], Peter VAN ROY[3], James CHAPMAN[1], Heinrich APFELMUS[5], Seyed Hossein HAERI[4,6], Magne HAVERAAEN[4], Mikhail BARASH[4], and Artjoms ŠINKAROVS[1]

[1] Input Output Global, UK
kevin.hammond@iohk.io, james.chapman@iohk.io, artjoms.sinkarovs@iohk.io

[2] Predictable Network Solutions Ltd. (PNSol), Stonehouse, UK
Peter.Thompson@pnsol.com, Neil.Davies@pnsol.com

[3] Université catholique de Louvain, Louvain-la-Neuve, Belgium
pvr@info.ucl.ac.be

[4] Department of Informatics, University of Bergen, Bergen, Norway
Magne.Haveraaen@uib.no, mikhail.barash@uib.no

[5] Cardano Stiftung, Zug, Switzerland
apfelmus@quantentunnel.de

[6] PLWorkz R&D, Belgium
hossein.haeri@gmail.com

Abstract. This paper describes the ΔQ systems design project. ΔQ is a methodology for design and diagnosis of complex distributed systems that has been applied at scale to a number of commercially successful projects in telecommunications, blockchain and other settings. It is supported by a tried and tested toolkit, written in Haskell.

Unlike other approaches, ΔQ directly captures key non-functional systems requirements in the form of composable outcomes, that can be used to construct statistical timing models, which exposes time as a first-class citizen. This allows overall systems feasibility to be determined *a priori*, and at low cost prior to major implementation being undertaken, as part of a structured design process. Where the methodology has not been used *a priori*, it also allows problems with timing (and other non-functional properties) to be diagnosed for systems that have been constructed and deployed, and to determine whether proposed solutions will enable the desired outcomes.

This paper introduces the ΔQ methodology, discusses how it has been applied in the context of Cardano, a top-10 blockchain that comprises around 900,000 lines of Haskell code, and discusses the supporting toolkit which is under development.

Keywords: Distributed Systems · Design-Time Feasibility · Cardano Blockchain · Timeliness · Process Calculi · System Design · ΔQSD · Functional Programming · Haskell

1 Introduction

Designing distributed systems to have predictable performance under all loads is difficult because the behaviour of system components and their interactions is both non-linear and stochastic. At high load, throughput and latency reach their limits and resources such as network capacity, CPU capacity, and storage will be exhausted, causing a dramatic effect on performance, particularly timeliness. Most existing approaches either fail to deal adequately with timeliness concerns, or else fail to capture timeliness properties in a way that permits the exploration of functionally equivalent distributed system designs. Fundamentally, what is needed is a way to expose timeliness properties (and other non-functional properties) as first-class citizens.

As part of long-standing work to predict and correct real-world problems with the performance of distributed systems [2,9,1,8,5,12], Predictable Network Solutions Ltd (PNSol) has led the development of the ΔQ Systems Development paradigm (ΔQSD) that aims to address this issue. ΔQSD has been used in areas as diverse as telecommunications [13,11,4,10], WiFi [7], and distributed ledgers [3,6]. It has been applied to many large industrial systems, with clients including BT, Vodafone, Boeing, Space and Defence, and IOG (formerly IOHK).

2 The ΔQ Methodology

In its broadest usage, ΔQ represents an approach to the modelling and analysis of distributed systems that treats timeliness as a first-class object, in contrast to most other approaches that foreground throughput and resource consumption, with timeliness as an afterthought. 'Timeliness' in this context includes failure, since from the perspective of the rest of the system, a component that fails is indistinguishable from one that does not deliver its output in time to be useful. More technically, then, ΔQ is a measure of timeliness that includes the possibility of failure. Existing systems can be measured and analysed with respect to ΔQ, for instance packet data networks can by analysed following [13]. Running systems can use real-time ΔQ measurements to adapt their behaviour, as in aspects of the Cardano blockchain data diffusion function [3,6].

ΔQ can also be applied a priori during system design by considering the system as a set of causally-connected outcomes. A design can be refined by decomposing an outcome into a collection of sub-outcomes: the ΔQ of the top-level outcome can then be calculated from the ΔQs of the sub-outcomes and their causal relationships. At any stage, the ΔQ of an outcome can simply be estimated; this then represents a timeliness budget that the subsequent refinement has to achieve. When it is clear that no plausible refinement can be expected to meet such a budget, the design is manifestly infeasible. This process allows design infeasibility to be detected at an early stage, before unfortunate decisions have been concretised into implementations that are expensive to alter.

2.1 Relationship to Functional Programming

While, as described above, the ΔQ methodology is generally applicable to complex distributed systems, it has deep roots in functional programming concepts and applications. In particular, it has been successfully applied to the Cardano blockchain node, which is written in Haskell, and which has been (partially) verified using Agda, QuickCheck, IOSimPOR and other lightweight formal methods. As described below, the methodology is supported by a new toolkit, which is written in Haskell.

3 The Cardano Blockchain

The ΔQ methodology forms an important part of the approach to lightweight formal methods that is used at IO Global to ensure rapid and effective development of high-assurance software, ensuring that important timeliness and other properties are considered throughout the software development process. In particular, it has been successfully deployed as a critical part of the high-assurance development methodology that has been used for Cardano.

Cardano is a fully distributed, permissionless blockchain that uses a proof-of-stake consensus algorithm called Ouroboros [**?**]. Nodes in the system that represent 'stake' take turns to mint blocks in the chain depending on a secret schedule. Because the identity of the next block minting node is unknown to the rest of the system, each newly minted block must be diffused with high probability to all potential minters in time for the next block to be minted. The timeliness of this diffusion process is thus central to the operation of the consensus algorithm.

4 The ΔQ Toolkit

A Haskell library has been developed to support the ΔQ system design methodology. This provides a set of functions to evaluate ΔQ expressions that represent the causal connections between outcomes, as formalised in [6]. These connections include:

- *Sequential composition*: an outcome is dependent on the completion of a previous outcome;
- *Last to finish*: two or more outcomes start concurrently, and the overall outcome completes when they have all completed;
- *First to finish*: two or more outcomes start concurrently, and the overall outcome completes when any one of them does;
- *Probabilistic choice*: one of two or more outcomes is selected with a given probability.

The library will be released as public open source on Hackage prior to the TFP 2025 meeting.

5 Proposed Talk Outline

1. Introduction to $\varDelta$Q: basic concepts including timeliness as a first-class citizen, outcomes, connections, outcome diagrams, feasibility.
2. Simple illustrative example: a small, distributed system.
3. The Cardano blockchain: timeliness requirements, $\varDelta$Q analysis.
4. The $\varDelta$Q toolkit: hands-on worked example (live coding).
5. Related approaches: process calculi, first-class time.
6. Conclusions: broader applicability, limitations, further work, exposing time a language property.

Acknowledgments. The work described in this paper was funded by Input Output Global.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Beuran, R., Ivanovici, M., Dobinson, B.: Network quality of service measurement system for application requirements evaluation. In: International Symposium on Performance Evaluation of Computer and Telecommunication Systems, SPECTS'03. pp. 380–387 (2003)
2. Bradley, J.T.: Towards Reliable Modelling with Stochastic Process Algebras. Ph.D. thesis, University of Bristol, Department of Computer Science (October 1999)
3. Coutts, D., Davies, N., Szamotulski, M., Thompson, P.: Introduction to the Design of the Data Diffusion and Networking for Cardano Shelley. Tech. rep., IOHK (August 2020), https://hydra.iohk.io/build/20405228/download/1/network-design.pdf
4. Davies, N., Thompson, P., Young, G., Newton, J., Teigen, B., Olden, M.: Measuring Network Impact on Application Outcomes Using Quality Attenuation. In: Measuring Network Quality for End-Users, pp. 43–52. Internet Architecture Board (2021), https://datatracker.ietf.org/meeting/interim-2021-mnqeuws-03/materials/slides-interim-2021-mnqeuws-03-sessa-neil-davies-peter-thompson-measuring-network-impact-on-appli
5. Gaixas, S.L., Perelló, J., Careglio, D., Grasa, E., Tarzan, M., Davies, N., Thompson, P.: Assuring QoS guarantees for heterogeneous services in RINA networks with $\varDelta$Q. In: 2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom). pp. 584–589 (2016). https://doi.org/10.1109/CloudCom.2016.0101
6. Haeri, S.H., Thompson, P., Davies, N., Van Roy, P., Hammond, K., Chapman, J.: Mind Your Outcomes: The $\varDelta$QSD Paradigm for Quality-Centric Systems Development and Its Application to a Blockchain Case Study. Computers **11**(3), 45 (2022). https://doi.org/10.3390/computers11030045, https://www.mdpi.com/2073-431X/11/3/45
7. Ivar Teigen, B., Davies, N., Olav Ellefsen, K., Skeie, T., Torresen, J.: Quantifying the Quality Attenuation of WiFi. In: Oteafy, S., Bulut, E., Tschorsch, F. (eds.) IEEE 47^{th} LCN. pp. 189–197. IEEE (2022). https://doi.org/10.1109/LCN53696.2022.9843690

8. Leahu, L.: Analysis and Predictive Modeling of the Performance of the ATLAS TDAQ Network. Ph.D. thesis, Bucharest, Tech. U. (Jan 2013)
9. Reeve, D.C.: A New Blueprint for Network QoS. Ph.D. thesis, Computing Laboratory, University of Kent, Canterbury, Kent, UK (August 2003), http://www.cs.kent.ac.uk/pubs/2003/1892
10. Teigen, B.I., Davies, N., Thompson, P., Ellefsen, K.O., Skeie, T., Torresen, J.: ΔQ Generative Models: Modeling Time-Variation in Network Quality. In: Charalambides, M., Papadimitriou, P., Cerroni, W., Kanhere, S.S., Mamatas, L. (eds.) 18^{th} CNSM. pp. 233–239. IEEE (Nov 2022). https://doi.org/10.23919/CNSM55787.2022.9965128, https://doi.org/10.23919/CNSM55787.2022.9965128
11. Thompson, P.: Tr-452.2 quality attenuation measurements using active test protocols. Tech. rep., The Broadband Forum (November 2022)
12. Thompson, P., Davies, N.: Towards a performance management architecture for large-scale distributed systems using RINA. In: 2020 23rd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN). pp. 29–34. Paris, France (2020)
13. Thompson, P., Hernandez, R.: Quality attenuation measurement architecture and requirements. Tech. Rep. TR-452.1, Broadband Forum (September 2020), https://www.broadband-forum.org/download/TR-452.1.pdf