# Towards Verification of Connection-Aware Transaction Models for Mobile Applications

Lars M. Kristensen[1] and Gabriele Taentzer[2] and Steffen Vaupel[2]

[1] Western Norway University of Applied Sciences
lmkr@hvl.no
[2] Phillips-Universität Marburg
{taentzer,svaupel}@informatik.uni-marburg.de

**Abstract.** Applications running on mobile devices are subject to frequent changes in connectivity to back-end infrastructure. In order not to disrupt service and ensure fault-tolerant operation, transaction-oriented mobile applications must be able to operate in both online and offline mode. Recently, a generic software architecture has been proposed [4] to accommodate mobile transaction models that support offline transaction processing in conjunction with data replication, reintegration, and synchronisation. We present an initial Coloured Petri Net (CPN) [2] model of a mobile transaction system and report on the first results on verifying its behavioural correctness using model checking.

**Introduction.** Mobile client applications often need to execute transactions that read and write shared data sets stored on a server-side infrastructure. Examples include applications involving local payment, where concurrently running applications need access to funds from a shared account. A challenge in this scenario is that mobile devices may often loose connectivity. To avoid disruption of service, the application must be able to operate even when the mobile device is *offline*. This requires specialised transaction models that replicate data for offline operation and which synchronise data when coming back *online*.

Several conflict-free transaction models have been proposed to support such scenarios. As an example, the Escrow transaction model [3] is based on a logical split of the shared data set, and can be used to for instance give a mobile application access to a restricted amount of funds on an account. Vaupel et al. [4] have proposed a generic architecture that includes online and offline transaction processing, replication, synchronisation and re-integration of data and which is able accommodate different conflict-free mobile transaction models.

**CPN model.** Our goal is to develop a formal executable specification of the mobile transaction architecture proposed in [4]. In particular, we want to verify the correctness of conflict-free transactions for a given mobile application. Furthermore, the CPN model should reflect the architecture and make it easy to change the set of transactions for a concrete mobile application.

Figure 1 shows the CPN module of the local transaction manager on the mobile client for an example with a Debit transaction operating on a shared

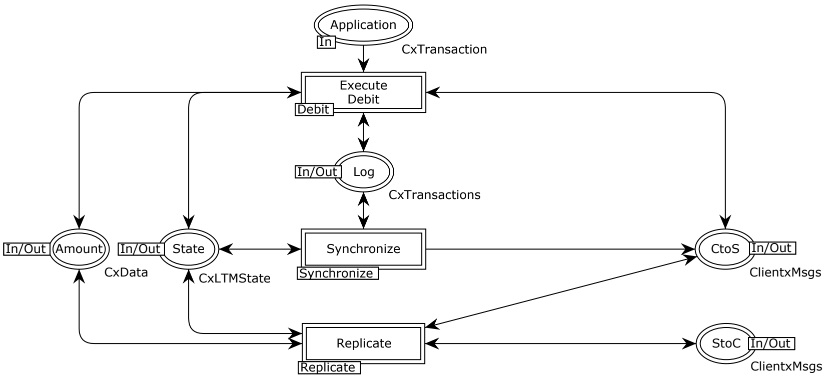

**Fig. 1.** Local Transaction Manager module.

Amount. The application invokes the debit transaction via the Application place. When operating in offline mode, the transactions executed are written in a Log. The substitution transitions Synchronise and Replicate represent the two major operational modes that allow data to be synchronised with the server-side when online, and conflict-free replication of data to support offline operation. The places CtoS and StoC are used for modelling the communication between the client-side and the server-side.

**Verification.** We perform verification using explicit-state model checking, as supported by CPN Tools [1]. The state space for the Escrow-based payment transaction system with a debit transaction has $7,174$ states and $22,202$ edges and can be generated in less than three seconds. The transaction model replicates the amount on the account such that all mobile clients have access to an equal amount. A key property of the application is that independently of how the clients go online and offline, it should always be possible to return to a *consistent state* in which the sum of the amounts replicated to the clients is equal to the total amount stored on the server-side. In computation tree logic (CTL), this property can be expressed as $AG\,EF\,p$, where p is a state predicate expressing that the state is consistent with respect to the amount.

## References

1. CPN Tools home page. `www.cpntools.org`.
2. K. Jensen and L.M. Kristensen. Coloured Petri Nets: A Graphical Language for Modelling and Validation of Concurrent Systems. *Communications of the ACM*, 58(6):61–70, 2015.
3. P. O'Neil. The Escrow Transactional Method. *ACM Transactions on Database Systems*, 11(4):405–430, 1986.
4. S. Vaupel, D. Wlochowitz, and G. Taentzer. A Generic Architecture Supporting Context-Aware Data and Transaction Management for Mobile Applications. In *Prof. of MobileSoft'16*, pages 111–122. ACM, 2016.