PEPERCORN: Inferring Performance Models from Location Tracking Data

Nikolas Anastasiou and William Knottenbelt

Department of Computing
Imperial College London
South Kensington Campus
London SW7 2AZ
{na405,wjk}@doc.ic.ac.uk

Abstract. Stochastic performance models are widely used to analyse the performance of systems that process customers and resources. However, the construction of such models is traditionally manual and therefore expensive, intrusive and prone to human error. In this paper we introduce PEPERCORN, a Petri Net Performance Model (PNPM) construction tool, which, given a dataset of raw location tracking traces obtained from a customer-processing system, automatically formulates and parameterises a corresponding Coloured Generalised Stochastic Petri Net (CGSPN) performance model.

Keywords: Performance Modelling, Location Tracking, Data Mining, Coloured Generalised Stochastic Petri Nets.

1 Introduction

Performance modelling and analysis facilitates the understanding of customer and resource flow in complex physical customer-processing systems, such as hospitals, airports and car assembly lines. The accurate formulation and parameterisation of a performance model is critical to the validity of subsequent analysis. Yet the construction of such a model usually requires the availability of large amounts of data. Current data-gathering techniques, such as time and motion studies, involve tedious manual tasks that are not only time consuming, but may also be inaccurate and disrupt the system's natural flow.

The increasing adoption of real time location systems (RTLSs) has led to an abundance of low-level data describing the fine-grained flow of customers and resources in customer-processing systems. In this paper, we introduce PEPERCORN, a tool which exploits the availability of such data in order to automatically infer and construct the PNPM of the underlying system and thus, provide insights regarding the system's high-level operations and performance. Constructed models can be visualised and/or analysed in PIPE2, the platform-independent Petri Net editor [4].

K. Joshi et al. (Eds.): QEST 2013, LNCS 8054, pp. 169–172, 2013.

2 PEPERCORN

PEPERCORN is a Java-based implementation of our earlier work [1–3] which presented a methodology, based on a four-stage data processing pipeline (cf. Figure 1), that allows the automated construction of CGSPN performance models from high-precision location tracking data. Key assumptions include static service areas with single-server service semantics and random service discipline.

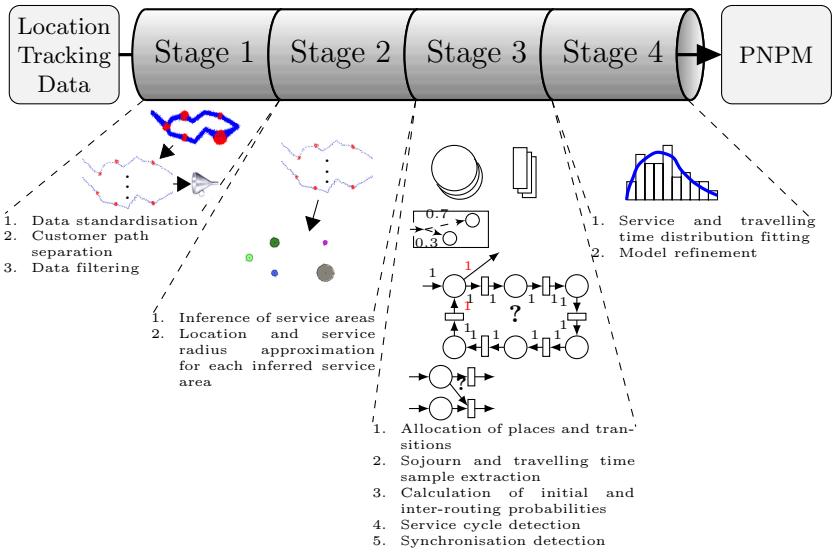

Fig. 1. The four-stage data processing pipeline that is implemented by PEPERCORN

The basic actions provided by PEPERCORN's main menu bar are shown in Figure 2. In order to construct a PNPM using PEPERCORN, the user first opens a file containing raw location tracking updates retrieved from a particular customer-processing system. PEPERCORN currently supports location tracking data obtained from a Ubisense UWB-based RTLS and synthetic data generated by the location-aware queueing network simulator LocTrackJINQS [6]. This data is a stream of tuples of the form `(tag id, type, x, y, time, stderr)`. The `tag id` field denotes the monitoring tag's unique identifier and `type` contains the tag's category, e.g. doctor, patient, etc. In the case of multiple customer classes, `type` can also be used to specify the customer class that the tag belongs to. `time` denotes the timestamp of the location update, i.e. the time when the location update was recorded by the RTLS, and `x`, `y` specify the location of the tag at that particular instance. `stderr` is the expected deviation between the tag's recorded location and actual location.

Once a file has been imported, the user can initiate the data processing pipeline through the 'Process File' toolbar button. The user may wish to adjust some of the pipeline's default parameters or disable the synchronisation

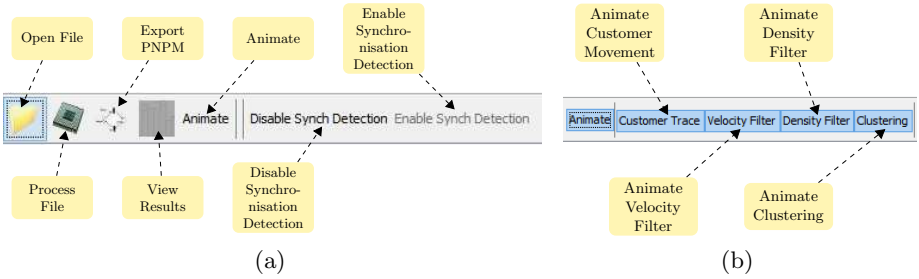

Fig. 2. Figure 2(a) shows the basic actions provided by PEPERCORN's toolbar. Figure 2(b) shows additional animation actions, once the 'Animate' button is pressed.

detection mechanism before processing commences. A user can also animate the processing phases of the second pipeline stage (cf. Figure 2(b)).

During data processing only one mandatory input is required: the *Eps* value used by the DBSCAN [5] clustering algorithm (cf. Figure 3(a)). This value defines the area of the neighbourhood around each point in the dataset for which the density is measured[1]. When processing is completed, the user can export the constructed PNPM as an XML file (a custom variation of PNML) so it can be visualised and/or analysed in PIPE2 (cf. Figure 3(b)). PEPERCORN also allows users to examine key quantitative results, such as the service and travelling time distribution fits (obtained by interfacing with the G-FIT tool [7]), the compatibility of the extracted time samples with the fitted distribution, and the inferred service area locations.

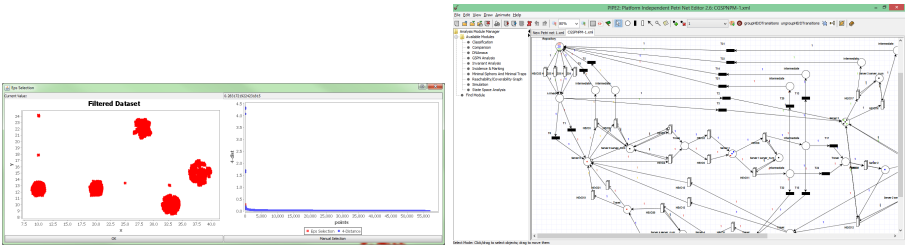

Fig. 3. Figure 3(a) shows the *Eps* selection dialog. Figure 3(b) shows the constructed PNPM as visualised in PIPE2 (in compact transition form).

[1] DBSCAN requires a second parameter in order to determine density-connected points: the value of the minimum number of points ($MinPts$) that must lie within the neighbourhood (defined by *Eps*) of each point in the dataset. However, we set the value of $MinPts$ to be equal to four in all cases after the suggestion of [5].

3 Conclusion

This paper has presented PEPERCORN, a tool used to automatically construct PNPMs by analysing the traces of the customer flow of customer-processing systems. This tool has been evaluated through a number of case studies in [1–3]. These case studies, conducted using synthetic location tracking data generated by LocTrackJINQS [6], employ several types of customer-processing systems, including systems with synchronisation, multiple customer classes and service cycles. Their results suggest that PEPERCORN is capable of inferring the abstract structure, stochastic features and high-level customer flow of complex systems, at least when synthetic location tracking data is used.

The constructed models can be used to provide insights into the system's performance through the computation of end-to-end response time distributions and to identify bottlenecks not likely to be discovered by a manual process. At this stage PEPERCORN is particularly suitable for small-scale indoor customer-processing systems characterised by complex processes which are difficult to capture via manually collected data.

References

1. Anastasiou, N., Horng, T.-C., Knottenbelt, W.: Deriving Generalised Stochastic Petri Net performance models from High-Precision Location Tracking Data. In: Proc. 5th Intl. Conference on Performance Evaluation Methodologies and Tools, VALUETOOLS 2011 (2011)
2. Anastasiou, N., Knottenbelt, W.: Deriving Coloured Generalised Stochastic Petri Net Performance Models from High-Precision Location Tracking Data. In: Proc. 4th ACM/SPEC International Conference on Performance Engineering (2013)
3. Anastasiou, N., Knottenbelt, W., Marin, A.: Automatic Synchronisation Detection in Petri Net Performance Models Derived from Location Tracking Data. In: Thomas, N. (ed.) EPEW 2011. LNCS, vol. 6977, pp. 29–41. Springer, Heidelberg (2011)
4. Dingle, N.J., Knottenbelt, W.J., Suto, T.: PIPE2: A tool for the Performance Evaluation of Generalised Stochastic Petri Nets. ACM SIGMETRICS Performance Evaluation Review 36(4), 34–39 (2009)
5. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proc. 2nd Intl. Conf. on Knowledge Discovery and Data Mining, KDD 1996 (1996)
6. Horng, T.-C., Anastasiou, N., Knottenbelt, W.: LocTrackJINQS: An Extensible Location-aware Simulation Tool for Multiclass Queueing Networks. In: Proc. 5th Intl. Workshop on Practical Applications of Stochastic Modelling (2011)
7. Thümmler, A., Buchholz, P., Telek, M.: A Novel Approach for Phase-Type Fitting with the EM Algorithm. IEEE Transactions on Dependable and Secure Computing 3, 245–258 (2005)