Workflow for the Internet of Things

Debnath Mukherjee, Debraj Pal and Prateep Misra
TCS Innovation Labs, Tata Consultancy Services Limited, New Town, Kolkata, India

Keywords: Workflow, Internet of Things, IoT, SLA, KPI, Prediction.

Abstract: Business Processes are an important part of a business. Businesses need to meet the SLA (Service Level Agreements) required by the customers. KPI (Key Performance Indicators) measure the efficiency and effectiveness of the business processes. Meeting SLA and improving the KPIs is the goal of an organization. In this paper, we describe the benefits of workflow technology for the IoT (Internet of Things) world. We discuss how workflows enable tracking of the state of various processes, thus giving the business owner an insight into the state of the business. We discuss how by defining IoT workflows, prediction of imminent violation of SLA can be achieved. We describe how IoT workflows can be triggered by the low level IoT messages. Finally, we show the architecture of an IoT workflow management system and present experimental results.

1 INTRODUCTION

Workflow Management Systems implement business processes which have both human participants and automated tasks. Workflows capture the state of a business process and enable state transitions when a trigger is received. An example of a trigger is: a workflow participant hits the "Submit" button on a loan approval form to approve a loan. Here the action of hitting the "Submit" button causes the state of the loan approval workflow to e.g. move from "Approve loan" to "Notify Loan Approval to Applicant". (A more formal definition of workflows is given in Section 3).

In this paper, we discuss the relevance of workflow technology in the context of the Internet of Things (IoT). Workflows used in IoT are termed in the rest of this paper as IoT workflow. In IoT workflows, there could be both human triggered state changes as well as triggers based on IoT messages. For example, in the case of package delivery by a courier, an IoT message "Package Received" may be generated when the recipient of a package signs on a hand-held device. This message is sent to an IoT cloud infrastructure which changes the state of the workflow to "Package delivered". Thus IoT workflows help in tracking the state of various processes in the enterprise.

Another important issue in a business is to keep track of Service Level Agreements (SLA) and Key Performance Indicators (KPIs). KPIs are more about organizational goals while an SLA is an agreement between the service provider and a customer. Example of a KPI could be "average cost of a package delivery trip". Example of SLA could be "latest time of delivery of package". We show in this paper that it is possible to keep track of imminent SLA violations (thus predicting the SLA violations) and take appropriate action.

The various advantages of IoT workflows are as mentioned below:

- IoT workflows will enable the users to get a view of the state of the business processes in real time
- It will enable the business to take action based on the data made available
- Compared to traditional workflows, the IoT workflows will give a granular view because it is based on the low level IoT messages
- Data about the business process and alerts (such as SLA violations) in the example can be stored and mined to derive insights
- Constantly monitoring the KPIs and SLAs will enable business benefits
- Predictions can be made in real time to avoid SLA violations

The key contributions of this paper are to explain how an IoT workflow system may be designed. It also describes how SLA violation prediction feature may be designed. It describes a scalability design for

Mukherjee, D., Pal, D. and Misra, P.
Workflow for the Internet of Things.
DOI: 10.5220/0006358607450751
In *Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS 2017) - Volume 2*, pages 745-751
ISBN: 978-989-758-248-6

the IoT workflow management system. Finally, it presents performance results.

The rest of the paper is organized as follows: Section 2 covers related work, Section 3 mentions some necessary definitions, Section 4 presents a motivating use case and lays down the design challenges, Section 5 describes the design, Experiments and Results are discussed in Section 6, and Section 7 presents conclusions and future work.

2 RELATED WORK

In this section, we discuss related work on SLA violations in workflows. Unfortunately, not much scholarly work exists on IoT workflows at the time of this writing.

(Emeakaroha, 2012) discusses the issue of maintaining customer specified SLAs in Cloud infrastructures. Cloud infrastructures need to be self-managed to minimize user intervention. This is achieved through timely detection of possible SLA violations. The paper describes an architecture (Detecting SLA Violation infrastructure (DeSVi)). It uses a framework which maps low level metrics (such as device uptime/downtime) to user specified SLAs. This helps to manage and prevent SLA violations. The DeSVi framework is validated in two applications: an image rendering application, and a web application running the TPC-W benchmark. SLA violation prediction is achieved through defining “threat thresholds” which are more restrictive SLAs than the SLAs themselves.

(Leitner, 2010) discusses the PREvent framework which monitors SLAs, predicts possible SLA violations using machine learning techniques, and takes necessary action to avoid the SLA violation. Prediction of SLA values is done at specific “checkpoints” using regression techniques (mostly using multilayer perceptrons). To avoid the predicted SLA violation, the framework uses an adaptation actions database. This database contains few actions that when applied singly or in combination can avoid the violation. For this, the framework knows in advance what is the impact of a specific action; this is done by estimating the improvement caused by a specific action – these are termed improvement estimates.

(Wetzstein, 2012) discusses a strategy for preventing KPI violations. Unlike (Leitner, 2010) it uses decision trees to model the relationship between low level metrics and higher level KPIs. The KPIs are predicted at specific points called checkpoints, similar to (Leitner, 2010). During the prediction, “instance trees” are derived which show which metrics need to be improved to reach specific KPI goals. Then adaptation requirements are identified based on the instance trees. Alternative adaptation strategies are ranked based on “preferences and constraints model” (Constraints must be met while preferences should be optimized). Finally, the adaptation strategy with the highest score is chosen for execution. In our work, the workflow is instrumented with domain specific code.

(Ivanovic, 2011) describes a scheme for SLA Violation prediction. It constructs a model of constraints that model both the states of SLA conformance and SLA violation. Based on this model it is able to predict when SLA may be violated.

Our work discusses the concept of workflows in IoT. Regarding SLA violation prediction, work such as (Leitner, 2010) differs from ours in that we instrument the workflow actions with prediction and alerting code. Actions taken upon alerting are work in progress, but the general direction (as explained in this paper) is towards a rule based diagnosis followed by a selection of best action.

3 DEFINITIONS

Most structured organizations have business processes modelled as workflows hosted in a workflow management system (WMS). Workflows are formally modelled as a WFNet (Workflow Net). In this section, we give some definitions regarding workflows. The definitions 1 through 3 are based on (Vander Aalst, 1998) while definitions 4 through 7 are added by this work.

Definition 1 (Petri Net) A Petri net is a triple (P,T,F) where:

- P is a finite set of places,

-T is a finite set of transition,

-F $\subseteq$ (P×T) $\cup$ (T×P) is a set of arcs

Places are conditions while transitions are tasks. Arcs connect a place to a transition or a transition to a place.

•t denotes the set of input places for a transition t. Similarly p• is a set of all transitions from an input place p.

Definition 2 (Strongly Connected) A Petri Net is said to be strongly connected if for every pair of nodes s and e there exists a path from s to e.(Nodes are either places or transitions).

Definition 3 (WFNet) A Petri Net WF is a WFNet iff:

- There are two special places i and o such that •i

= ϕ and o• = ϕ (Note : i is the start node and o is the end node).

- If a transition τ is added between o and i (i.e. such that o• = { τ } and •i = { τ }), then the WF is strongly connected

Definition 4 (Trigger) A trigger T is a message sent to a Workflow Management System, W which causes the workflow to change state from its current state at time t, S_t ,to state S_{t+1} at time (t+1) , where S_t and S_{t+1} are transitions of the WFNet.

Definition 5 (IoT message) A message M is an IoT message if it is generated by a sensor, S, and sent to the IoT platform, P.

Definition 6 (IoT message trigger) A trigger T is said to be an IoT message trigger if T is an IoT message and is a trigger.

Definition 7 (IoT workflow) An IoT workflow is a WFNet where the set of triggers which causes its state transitions must include at least one IoT message trigger.

4 MOTIVATING USE CASE

Consider a courier company, Delivery Express, which delivers packages every day. Each package has to be delivered within a certain time – we call this the Service Level Agreement. The delivery trucks leave the package store in the morning and deliver multiple packages along some routes. The state of the package deliveries is tracked by a workflow as depicted in Figure 1.

In the package delivery example, the state transitions can be triggered manually by the delivery personnel using e.g. a mobile application or can be inferred from the GPS messages. There could also be IoT messages triggering the workflows – e.g. a signature on a handheld device can send a "Package Delivered" IoT message to the workflow management system.

Predictions have to be computed at each step. For example, before step T4 starts (or even earlier), it should be known if Package B's SLA will be violated, so that appropriate actions can be taken.

The key design issues to be handled are as follows:

- How to model an IoT message triggering a workflow state transition, and
- How to implement the SLA violation prediction feature.

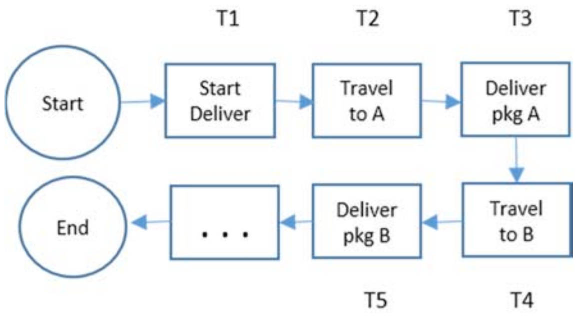

Figure 1: Package Delivery workflow.

5 DESIGN

In this section, we discuss the key design issues: handling IoT message in triggering workflows, and implementing prediction of SLA violation.

5.1 Handling IoT Messages in Workflows

One of the main features of IoT workflows is their ability to support IoT messages. The "things" send messages to the cloud-hosted workflow engine. These messages trigger state transition in the workflows. One key difference between the standard business workflows and the IoT workflows is the ability of the latter to bridge the cyber-physical divide. With the "things" now equipped with sensors, their messages can be monitored thus enabling their states to be tracked.

Our prototype IoT workflow system is WFMS. The WFMS workflows support different types of nodes (equivalent to transitions in WFNet). Different types of nodes include "User Task" which handles inputs from a human operator, and "Signal" which waits for a signal of a particular type.

The "Signal" node can be used to support IoT messages. To specify a Signal node, a signal "event type" has to be specified by the "thing". The message payload can also be sent to the signal node which can be used by the workflow in its information processing in later steps. An architecture diagram for the IoT message handling is shown in Figure 2.

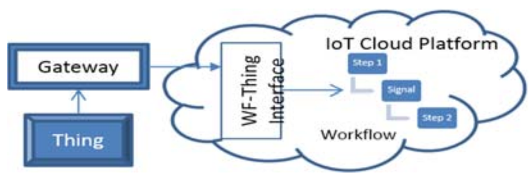

Figure 2: Handling IoT messages in a workflow.

The architecture shown in Figure 2 depicts the “Thing” sending sensor messages to a gateway which transmits to the IoT Cloud Platform. The sensor message is routed to a “WF-Thing Interface” which extracts the message payload and the event type and signals the workflow to change its state. Note that the workflow will wait until the message of the correct event type is sent to it.

One of the key design issues to be kept in mind is how to integrate the workflow system with the IoT messaging system. The IoT system sends messages which have the following fields: <asset id> <asset property> <current state> <message>. The asset id identifies the asset (or entity) that is sending the message. Asset property is the property of the asset whose value has been changed as a result of an event in the IoT world. The current state is the state of the workflow that is known to the IoT event producer. For example, for a user task, the event would be a form submission by a user. The form contains the current state of the workflow (it is passed onto the form by the workflow system). The combination (asset id, asset property, current state) may be sufficient to identify the process instance that needs to be triggered. As an example, the asset may be the mobile phone of the driver of the courier delivery truck, and its property may be an information field that is associated with the workflow client mobile or web application. Once the process instance is identified, it is triggered based on the message field – this field contains the type of event and this is compared with the event(s) that the process instance is waiting for. After this the workflow steps are run and then the workflow waits for another event.

5.2 Detection and Prediction of SLA Violation

In addition to providing tracking abilities, IoT workflows can provide insights into the state of the process. In particular it may be used to detect whether an SLA violation has happened or is about to happen.

At every step, or periodically (or continuously if feasible), forecasting needs to be done to determine if any KPI or SLA would be impacted. A forecast of the parameters that could affect the SLA or KPI is done. If it is so determined that a KPI or SLA is affected then corrective measures would be needed to address the threat.

For example the diagnosis may be that there was a traffic jam and the best action could be to re-route the vehicle.

Thus the predictions module goes through the following methodology:

- Forecast the future values of the parameters that affect KPIs / SLAs
- Detect threat to KPI / SLA
- Diagnose possible cause(s)
- Action after evaluating all possible options

An implementation of the prediction may be done by instrumenting the IoT workflow with code that predicts the SLA violation. The instrumentation can be put into separate nodes (e.g. in “Script Tasks” of BPMN).

In the package delivery example, the expected time to travel and deliver packages is known based on both internal and external knowledge. For example, the typical package delivery time (once the delivery truck has reached a destination) can be known from the historical records of the courier company. The expected time to travel can be known from some external data providers like Google maps. Expected time to complete various steps of the workflow may be revised in real time based on current situation, such as current traffic conditions.

The workflow also predicts whether the SLA for package B is going to be violated. This could be done through a check at the end of T2: ET(T2) + E(T3) + E(T4) > SLA(B) (it could, in more complex implementations, call a traffic prediction service), where ET(X) is the end time of node X and E(X) is the expected time for node X. For predicted SLA violations the system issues an alert and gets further guidance (such as “change routes” etc.).

Let us say that we have to predict the expected time to complete item X which is introduced at step *start* at time t_s. X is to be completed in step *finish*. Then the algorithm Predict_SLA_Violation predicts whether the SLA of X, SLA(X) will be violated. The prediction is computed at step s’ (which occurs before finish and starts at $t_{s'}$)

Algorithm *Predict_SLA_Violation (item X, step s’)*

1) The predicted time of completion of X, p, is computed as:

$$p = \sum_{si \in path(s', finish)} E(si) + \text{ts}'$$

where path(x,y) is the set of places and transitions (or nodes) that lie along the path from node x to node y.

2) For parallel paths existing between s’ and *finish*, for each parallel path, the expected time would be the expected time of the

longest (slowest) path amongst the parallel branches.

3) If p > SLA(X), then SLA will be violated, otherwise not

One assumption in the above algorithm is that the path selection decided in the decision nodes is known beforehand, in case decision nodes are present in the workflow.

5.3 The Workflow Management System Design

A workflow management system, WFMS, has been designed and implemented. Figure 3 shows the main components of WFMS. It implements the BPMN standard. IoT messages are handled through the "intermediate catch event" nodes which wait for IoT messages. Broadly the design includes the two phases: parsing and execution. During parsing, the process definition specified in BPMN XML format is parsed and converted to Java objects. During execution, state changes are performed based on IoT messages received. In the case of human tasks ("User Task" in BPMN), the workflow transitions from the sub-states :"initiated" (when the user task is created) to "claimed" (when a BPMN "potential owner" claims the task) to "started" (when a user starts work on the user task) and finally "completed", when the work is done and the different variable values are submitted.

The workflow management system is currently available as a web service (implementing RESTful web services). The different services are: "create process definition", where the BPMN XML process definition is input along with process name and version, the "initiate process instance" service where the user inputs the process name and version and an active instance of the process is created and the process instance identifier is returned, the "list all active instances" service where all active instances with their process instance identifier, process name/version and current state are returned. Finally an "interact with process instance" service is offered

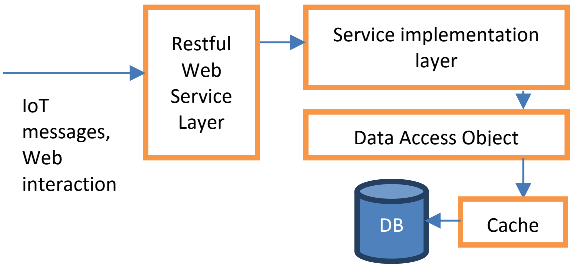

Figure 3: Architecture of WFMS.

which offers the following: a facility to send a signal (event) to a process by providing as input the process instance identifier, and the signal name, and a facility to inspect the values of variables belonging to a process instance. The WFMS also supports Java based script tasks as automated activities.

The architecture of WFMS is shown in Figure 3. The RESTful web services are implemented in the Service Implementation layer. The persistence is handled through Data Access Object layer and a cache layer. At the present time, the database interactions are not implemented and the WFMS is an in-memory workflow management system.

5.4 Scalability Considerations

As there is expected to be a large number of users of the workflow management system (henceforth referred to as WMS), it is important for the system to be scalable.

The system on which the WMS was targeted to be deployed was a multi-core processor. So the natural choice was to have the scalable system implemented as a multi-threaded application. Each process was assigned to a thread and all activities of a workflow process instance were done in that thread. To distribute the computation, there would be a load balancer which equitably balances the state change activities amongst the threads. We used round-robin load balancing amongst the threads. Whenever a request comes, it is mapped to the thread handling the process associated with the request. The scalable architecture is shown in Figure 4.

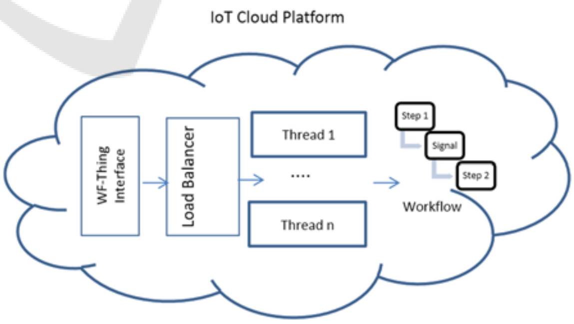

Figure 4: Architecture for scalability.

Another option would be to have each thread handle activities of multiple processes distributed equitably amongst the threads. In addition to distributing the processing via threads, distribution across machines (e.g. using a cluster of machines) is possible.

5.5 How IoT Workflow Advantages Are Achieved

We describe in this section how IoT workflows achieve the advantages mentioned in the introduction section.

To enable the business process owners to track the processes, a dashboard containing the state of each business process along with the relevant KPIs and SLAs can be provided.

Based on the SLA violation alerts, the system can take action and can diagnose the possible cause of the alert. For example, the possible causes of an SLA violation could be some malfunction within the delivery vehicle (determined from the on board diagnostics in the vehicle) or traffic congestions along the route (determined from traffic feeds). Rules can be designed considering the above causes, and these rules trigger actions such as a command sent to the vehicle operator to change routes.

IoT workflows enable the tracking of granular activities. Since the IoT "things" are provided with sensors, granular monitoring of activities is possible.

IoT workflows generate data about processes that can be mined. For example, the SLA violation alert data can be a useful source of insights. In the courier example, it may be found that often SLAs were violated when the vehicle was travelling on a particular route segment. The action would be to avoid the route segment, if feasible, or start earlier.

Constant monitoring (not missing a single alert) helps the business stay on course to meet its KPIs.

Real time predictions can be very useful in dynamic situations. In the courier example, if an SLA is predicted to be violated, then a real time action such as changing routes to avoid traffic, can actually avoid the SLA violation.

6 EXPERIMENTS AND RESULTS

Since the IoT platform is used by multiple users, it is expected that a large number of workflows will be simultaneously active in the system. A large number of state transitions and automated tasks will be scheduled on the workflow management system. Hence it is important to measure the performance of the workflow engine.

The package delivery workflow was executed for a number of processes. In an iterative (sequential) fashion, each task of each process was triggered (the first task of all the processes were executed, then the second task and so on). The results for the JBPM workflow engine (an open source workflow system) and WFMS are as shown in Table 1.

Table 1: Performance for multiple processes.

Number of processes	Time taken in JBPM(seconds)	Time taken in WFMS(seconds)
100	10.923	3.013
200	20.469	3.168
300	35.476	3.200
400	44.369	3.278
500	61.888	3.293

The better performance of WFMS could be due to efficient data structures such as Maps used as well as due to the fact that JBPM uses an in-memory database whereas WFMS uses in-memory data structures.

7 CONCLUSION AND FUTURE WORK

In this paper, we have shown that workflows are important in the context of IoT. Workflows help in tracking of processes and also provide insights into the business – such as whether there are any impacts to the SLAs or KPIs. An implementation of IoT workflow, which supports receiving IoT messages and instrumentation of processes to predict SLA violations, has been evaluated in experiments.

IoT workflows allow business users real time visibility of the state of processes and enable businesses to take action based on the data. IoT workflows give a detailed view of the processes by bridging the cyber-physical divide using sensor data.

In future we wish to implement scalability related enhancements (see Section 5.4) and measure the scalability of the workflow engine. We will add a persistent data store for storing workflow state. Also, a library of standard prediction methodologies will be provided along with the workflow.

REFERENCES

Emeakaroha, V. C., Netto, M. A. S., Calheiros, R. N., Brandic, I., Buyya, R., De Rose, C. A. F.Moore, R., Lopes, J., 2012. Towards autonomic detection of SLA violations in Cloud infrastructures. In *Future Generation Computer Systems*. Elsevier Science Publishers.

Ivanovic, D., Carro, M., Hermenegildo, M., 2011. Constraint-Based Runtime Prediction of SLA Violations in Service Orchestrations. In *ICSOC'2011,* *International Conference on Service-Oriented Computing.* Springer-Verlag Berlin, Heidelberg.

Wetzstein, B. et al. 2012. Preventing KPI Violations in Business Processes based on Decision Tree Learning and Proactive Runtime Adaptation. In *Journal of Systems Integration*.

Leitner, P., Michlmayr, A., Rosenberg, F., Dustdar, S. 2010. Monitoring, Prediction and Prevention of SLA Violations in Composite Services. In *IEEE International Conference on Web Services*. IEEE Computer Society.

Van der Aalst, W. M. P. 1998. The Application of Petri Nets to Workflow Management. In *Journal of Circuits, Systems and Computers*. World Scientific.