Resilience Analysis of Collaborative Process Management Systems

Paul De Vrieze, Lai Xu

Resilience Analysis of Collaborative Process Management Systems

Paul de Vrieze and Lai Xu
Faculty of Science & Technology
Bournemouth University, Poole House, Talbot Campus,
Fern Barrow, Poole, BH12 5BB, UK
{pdvrieze, lxu}@bournemouth.ac.uk

Abstract. Process management allows for the automated coordination of processes involving human and computer actors. In modern economies it is increasingly needed for this coordination to be not only within organizations but also to cross organizational boundaries. The dependence on the performance of other organizations should however be limited, and the control over the own processes is required from a competitiveness perspective. Overall, this indicates a federated process management approach instead of a centralized one. This paper analyses the resilience of automated process management overall and especially how the use of federated process management impacts that resilience.

Keywords: Collaborative process systems, resilience, federated process management, process resilience

1 Introduction

Collaborative networks as a relative new scientific discipline [13] have been applied to application areas such as Factories of the Future [14], Manufacturing and logistics networks [15]. Organizations, enterprises, and communities are interconnected by networks in the new application areas. To support such collaboration in a hyper-connected world, existing technologies need to be improved and adapted in terms of larger scale integration and more intelligent devices, sensors and cyber-physical systems involvement.

The scope of business collaborations can be limited to a particular department in an enterprise, it may span multiple divisions within an enterprise or it may require interenterprise collaboration. In this research, we are especially considering how to manage interenterprise collaboration, i.e. collaborative business processes are executed among enterprises. In this work, we consider that (web) services and human interactions are coordinated through an automated process management system to form an SOA. In a traditional approach, the process management system forms a single point of failure. We defining resilience as the degree as to which the system can continue to meet its goals in the presence of errors. Based upon an analysis of different structures for federation we have analyzed how process federation can impact overall system resilience.

In this paper we focus on federated control of collaborative business process systems as well as analysis of resilience related the federated collaborative business process systems. Federation in this context is the coordination of the execution of individual business process instances by multiple independent process engines. In the next section, we review challenges of collaborative networks and collaborative business process managements, federated control of information systems, large-scale systems, business process management systems/business services, and cloud services. In Section 3, different management types of federated collaborative processes are classified. Resilience of the federated collaborative processes is reviewed in Section 4. The paper is end with a summary of the paper and future research.

2 Related Work

Increasing importance of value chains and production networks, of interconnected organizations, collaboration dynamics, outsourcing, and the increasing potential of new ICT technologies supported innovations has driven research of collaborative networks [12]. Collaborative networks [6] [13] and collaborative business process management [11] [12] have been fostered by globalization over decades.

Federated control method has been applied into different areas. Federated information management architecture is provided to manage information across different organizations in virtual organization domain [3]. Different management level schemas of federated databases are specified for interoperation.

Systems control is another research area which federated control methods has been applied [6]. In this area, large-scale systems normally consist of several interconnected local systems, which conventional centralized control schemes are not suitable for due to global dynamic behavior complexity and computational difficulties. Many multi-agent based federated control systems are used to high performance computing area. Federated Cloud Services [8] [9] [10] emerges recently to deploy and to manage multiple external and internal cloud computing services for meeting business needs.

Paper [16] proposes a framework to create enterprise resilience using service oriented architecture approach. Antunes and Mourão consider a framework and related services for building resilient business process management [17]. Both works are not look at resilience of collaborative systems. Previous work on federated business process management [1] [2] [5] are proposed to handle the dynamic nature of collaborations among autonomous enterprises, which different structures are specified for different interactions among business partners.

Today we are facing to support enterprises' agility and resilience in turbulent business environments [4]. Advances in ICT, Internet, and cloud have led to the explosion of collaborative networks both people and organizational levels. A pervasive interconnection of and collaboration among physical and virtual objects provides opportunities of realization of manufacturing 2.0 (industrial 4.0) and sustainable development. Our previous work [7] provides a method to analyze resilience of a cloud-based collaborative process solution. We thus review federated collaborative process management and analyzing resilience of federated collaborative process management in the new context.

3 Federated Collaborative Process Systems

Processes that involve multiple parties may be collaborative. While it is possible for this to be limited to two parties, generally one would expect more than two parties. Various degrees of collaborativeness can be distinguished. At the lowest level of collaborativeness, collaborative processes have little differences to distinguish them from simple provision of (web) services. At the highest end there are complex interactions between the involved parties.

In addition to degrees of collaborativeness, there is the dimension of balance of power or control. Collaborations with dominant partners, as for example occurs in supplier networks for large manufacturers of cars or airplanes, are quite different from more peer-to-peer oriented collaborations where no partner is dominant.

3.1 Collaborative Processes

In terms of process management, it is convenient to make the distinction between service provision/consumption on the one hand and collaboration on the other hand. Starting from the precept that all processes have a single initiator, a process can be seen as collaborative if, at operational level (not purely on an intellectual basis with human owners) the initiator and executor have some awareness of each other's process and use this in some aspects of operations. The use of process nature can be limited, for example to monitoring, but there should be some use of the process nature.

In real terms we primarily consider automated processes that are managed through some form of workflow or business process management system. In this digital representation a process is represented as control flow as well as activities whose interaction is managed in line with the control flow to provide the process outcomes. Please note, that in this context we look at logical uniqueness and do not consider load balancing or replication of services or process management systems.

Centralised Process Management Traditionally process choreography is seen as performed by a single process management system (or process engine). This system has a full view of the process and all components. In case that external components are involved, they are treated opaquely as service consumption where the system itself has no further knowledge of the underlying processes or possible interactions.

In this scenario, collaborative process management would imply that the management system requires quite some insight into and control over the implementation of activities within the collaborating organizations. While cloud computing makes this somewhat easier, many collaborators may object to this. In most cases, tight contractual arrangements are needed to allow this to happen [11] [12], and this is mainly seen in a context with dominant players.

3.2 Federated Process Management

Federated processes are distinguished from centralized process management in that process-aware collaboration occurs, but is controlled by multiple autonomous

management entities. While federated process execution is normal when processes are managed and executed by humans (each person manages his own actions within the agreed boundaries), such coordination is not the norm for automated process management.

Federation as concept implies that there is some degree of meta-level coordination between the management agents that perform the service choreographies. Considering multiple copies of the same process (description) to be the same process (model), multiple instances of a single process model can, from a technical perspective, be instantiated on/coordinated by different management systems. For overall management it is however worthwhile that meta-information is coordinated to allow for overall statistics. Each process instance should have a single, eventually consistent, state.

When parts of a process are coordinated by different process management systems in a federation, there are different ways this can be organized. In this paper we only consider those approaches where at any point of time there is a single "master" or "owner" of the process instance. The coordination of a subset of the activities can be delegated to a different management engine. Ownership may also be transferred (voluntarily or in response to failure). While many real-world process would consist of combinations of these (descriptive) classifications, it is still worthwhile to consider them separately.

Hierarchical federated processes Within federated process management, some processes are purely hierarchical. This means that the process is designed such that process instances are primarily managed by a single management system that delegates parts of the execution to other process management systems under control of a different collaborator. If we define an execution domain as those parts of a process managed and choreographed by a single management system, hierarchical federation is defined through a strict hierarchy of execution domains.

An example of a hierarchical federated process setup could be where a cooperative of specialized manufacturing companies is set up as collaborative network organization to be able to deliver more end-to-end solutions. As part of this collaborative network some centralized coordination is required and can be provided and managed by the cooperative. Individual manufacturers would however retain control over their own processes, but the overall process would be managed by the master process management system.

Sequential federated processes A sequential federated process is a process where there is a clear transition from one locus of control to another, after which the original execution domain is no longer involved in the choreography of the process. Effectively this means that responsibility and control can be transferred.

Free-form federated processes In many cases, a strict hierarchy of execution domains is a significant restriction and sequential federation is also not applicable. Free-form federated processes as such are those processes where there are no restrictions are set on the relations between execution domains.

4 Collaborative Network Resilience

In determining resilience of a process system within a collaborative network it is first of all of import to define what degrees of resilience are considered. In this sense we only consider the resilience of the system, not the resilience of the network as a whole.

Resilience of a process system exists on four levels: the resilience of the system *overall*, the resilience of *individual processes* within the system, the resilience of *process instances* and the resilience of individual constituent *services*. In achieving resilience one would certainly require that failures at a lower level have minimal impact on the higher layers, but the failure forms do inform the achievement of resilience at a higher layer.

In this paper, resilience of the federated collaborative process systems can be defined as adaptive capacity and its ability to cope with, adapt to and recover after a disruption or failure. Thus we first look at failure model in Section 4.1 as well as service resilience (Section 4.2), process instance resilience (Section 4.3), process resilience (Section 4.4) and overall resilience (Section 4.5) respectively.

4.1 Failure Modes

The various process system aspects can fail in various ways, but they share many aspects of the ways in which they can fail. For the purposes of resilience, only the external aspects of failures are of interest. We are also looking at failure from a software perspective, where manufacturing failure modes inform the perspective taken, but are not sufficient to analyze interactions across levels of resilience or the kinds of systems. Overall we distinguish the following orthogonal dimensions (please note that these are not exclusive):

- **Temporal failure**: Along the time dimension a failure can be temporary (an independent temporary event, for example a network failure due to a power cut); failures can be intermittent based on a systematic issue (related failures for example due to insufficient capacity during nightly batch operations); Failures can also be permanent (the company providing the service has seized operation).
- **Failure detectability**: Some failures are easy to recognize such as those where error messages are provided, *signaled failure*. Other failures are detectable based on business rules or SLA's such as failure to respond within a certain time period,

detectable failure. The final category of failure detection is are silent failures either in silent non-performance (ie. the request is accepted but never actioned), *silent partial-performance* (ie. some aspects of the request are not performed, but those aspects performed are valid. The final form of silent failure is *silent corruption* where the system provides incorrect state or results without signaling any error.

- **Partial failure**: Partial failure concerns the degree to which the system is failing. Full failure is a possibility, but partial failure can occur in various forms such as *degraded performance* (eg. not meeting time expectations), *reduced quality* (the results are valid but not of the quality or precision normally provided, the range of the function is reduced) or *reduced provision* (the domain or image of the functionality is reduced, ie. not all inputs are supported, for example a weather service would have reduced provision for an area if the weather station in that area was unavailable for some reason).
- **Failure origin**: The origin of a failure can either be *internal* as in it is caused by issues within the logic control of a system, or *external* where the failure is outside of the control of the system. An internal failure could be a bug in a service implementation or process description. A bug in the process management system would be external from the perspective of a process. A network failure that causes a service to be unavailable would be external to that service. Both examples can, in circumstances, be seen as internal to the overall system. An internal failure can either be a *delegate failure* (a used component causes the problem) or a *direct failure* where the problem is not fundamentally in the delegate (although a delegate may return unexpected (but not invalid) results that trigger the issue).

4.2 Service Resilience

Services form the building blocks of a SOA based process system. The causes of their failures are therefore out of scope of this discussion, but they may exhibit all forms of these failures. These failures do impact the overarching resilience levels however as for the system to work, the failures need to be able to be handled.

4.3 Process Instance resilience

Process instance resilience relates to how process instances can recover from failures. In this aspect, direct failures are normally due to incorrect assumptions or pure logic errors (bugs in the process model), and of less interest within the resilience discussion.

From a temporal dimension perspective, process instances are, due to their inherent nature as long-running, reasonably resilient to non-permanent failures. The lower the pressure on timing, the stronger the resilience.

From a detectability perspective, only signaled failures are normally handled by process instances, and higher resilience generally implies more complete failure management, that includes the availability of sufficient semantic information such as to allow automatic restarts or rerouting for failed paths/components. Silent non-

performance can possibly be handled through monitoring and timeouts, but in many cases cannot be distinguished from other forms of silent failure.

In terms of partial failure, degraded performance may lead to timing issues on the process instance, but otherwise should not impact the performance of the process. Reduced quality of performance leads to reduced quality of the instance, and reduced provision would lead to some instances failing (those affected by the reduction) and some succeeding.

In relation to the location of failure, external failure is normally failure of delegates/activities, but can also be a failure in the execution environment (eg. hardware failure). External failure of the execution environment could be compensated by regular replication solutions for the state of the execution environment (eg. database replication / sharding etc.) that is not specific to process management. In terms of direct failure, there are two aspects to the resilience of process models. First is the ability of humans to interject in process instances and possibly modify them or provide "missing" data, this mainly a property of the management system and the used process language. The other aspect is the frequency of instances. In other words, while it may be technically feasible to intervene in a process instance and recover it, it may not be feasible for large amounts of instances that failed due to fundamentally the same cause.

4.4 Process Resilience

The resilience of processes is strongly related to the resilience of process instances. There are however ways in which the execution system can make processes be resilient even when individual instances fail. Conversely, timing and speed issues that are not problematic at instance level, may become so at process level.

Process resilience at a temporal level requires monitoring and (pre-emptive) intervention and/or rerouting. Processes are resilient when it is easy to (automatically) replace the providers of services for the process (activity execution or process management), but it requires a resilient system to actually make use of those possibilities. Obviously if these measures can be performed on already started process instances, this would even be better.

At process level, the detectability of failures is limited to the ability to express overall expectations, possibly at aggregate level. These process level specifications could then allow for pre-emptive intervention in the instantiation of new processes. Note that at process or system level it may also be feasible to determine some degree of likelihood of the various forms of silent failure by aggregating monitoring data and linking it to the expectations.

At process level partial failure of some instances would constitute partial failure of the process. A resilient process would allow the instance failures to be minimized and possibly corrected through re-execution along an alternative path. To do the latter, does require sufficient information to be available for the process components to be able to determine the semantic validity of doing so.

Locus of failure at process level implies some degree of process quality requirements as well as designs that minimize the impact of external failures. Again, monitoring of individual service and instance performance can help in detecting, mitigating and avoiding these failures or transitioning them from non-performance to

degraded performance by for example choosing a lower quality service that does not have availability issues.

4.5 Overall Resilience

Overall the entire systems resilience is a combination of the resilience of the constituting processes. Many measures at overall level can however help to achieve this resilience. In particular the provision of fallback execution environments or services will improve the overall resilience of the system. Resilience in terms of fault tolerance mainly depends on the resilience of the components (services, processes) and sufficient provision of execution environments without single point of failure.

5 Resilience for Federated Process Management

Where the previous section discussed the resilience of the components that make up a federated process management system that is on a technical level rather than a utility level. The possibilities of federated process management can, on a theoretical level be seen as a subset of what single entity-controlled process management provides. on a practical level, there are resilience benefits to federated process execution (there are also disadvantages in complexity and management). In addition, in real-world business scenarios it is often not desirable to have centralized process control.

In terms of resilience, federated process management means that by implication, processes are managed by multiple independent execution environments. While these environments may have access to different components (such as company internal services) they would have similar orchestration capabilities. As federated processes allow parts of execution to be delegated to other execution environments this means that an overall master coordinator does not require direct access to all required components for process execution.

At system and process level, through a lack of single point of failure and lack of global coordination requirement, federation provides a strong level of resilience against total failure. Performance degradation, especially non-functional degradation, can be compensated for by additional provisioning elsewhere within the collaboration. Even the absence of certain unique resources, would only result in those processes being unavailable.

The services used by a system are independent of the form of process management, and their resilience is mainly a property that influences higher levels. Federation however brings some interesting perspectives to process instance resilience. In the case of sequential federation, given a well-defined ownership transition model, the failure of preceding execution environments would not lead to instance failure.

Within hierarchical federation it is possible that, beside the master execution environment multiple delegate environments are active concurrently. Fallback, if semantically valid within the process, would however still be possible as long as a strict order of fallbacks is known by all candidates. These candidates would also be

required to have some degree of overall knowledge of the process, possibly under some sort of escrow approach.

6 Conclusion

From the discussion above, it is clear that beyond meeting business needs in providing the benefits of collaboration while safeguarding independence of control, federated process environments also bring interesting resilience properties. In particular, the independence of execution environments prevents the full failure at system level in the case of failure of individual execution environments. At process level, full failure is limited to those cases where no fallback to a component is available (a unique, mandatory, service only provided by a single partner). Instance resilience is no longer restricted to an ability to work around failure of single components, but also handle failure of execution environments.

Process resilience, in federated and other forms, does however require processes to be designed for resilience, to have additional information that specifies the semantic validity of cancellation and retries, and for there to be good, shared, monitoring of component performance and reliability.

References

1. Kutvonen L, Ruokolainen T, Metso J. Interoperability middleware for federated business services in web-Pilarcos. Agent and Web Service Technologies in Virtual Enterprises. 2007 Jul 31:288.
2. Kutvonen L, Metso J, Ruokolainen T. Inter-enterprise collaboration management in dynamic business networks. In On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE 2005 Oct 31 (pp. 593-611). Springer Berlin Heidelberg.
3. Afsarmanesh H, Camarinha-Matos LM. Federated information management for cooperative virtual organizations. In Database and Expert Systems Applications 1997 Sep 1 (pp. 561-572). Springer Berlin Heidelberg.
4. Camarinha-Matos LM. Collaborative networks: A mechanism for enterprise agility and resilience. In Enterprise interoperability VI 2014 (pp. 3-11). Springer International Publishing.
5. Leymann F, Roller D, Schmidt MT. Web services and business process management. IBM systems Journal. 2002;41(2):198-211.
6. Dong Q, Bradshaw K, Ferrese F, Bai L, Biswas S. Cooperative Federated Multi-Agent Control of Large Scale Systems. In ACTA Control and Applications Conference 2011 Jun.
7. de Vrieze P, Xu L. An Analysis of Resilience of a Cloud Based Incident Notification Process. In Risks and Resilience of Collaborative Networks 2015 Oct 5 (pp. 110-121). Springer International Publishing.
8. Buyya R, Ranjan R, Calheiros RN. Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services. InAlgorithms and architectures for parallel processing 2010 May 21 (pp. 13-31). Springer Berlin Heidelberg..
9. Demchenko Y, Ngo C, de Laat C, Lee CK. Federated access control in heterogeneous intercloud environment: basic models and architecture patterns. InCloud Engineering (IC2E), 2014 IEEE International Conference on 2014 Mar 11 (pp. 439-445). IEEE.

10. Konstanteli K, Cucinotta T, Psychas K, Varvarigou TA. Elastic admission control for federated cloud services. Cloud Computing, IEEE Transactions on. 2014 Jul 1;2(3):348-61.
11. Liu C, Li Q, Zhao X. Challenges and opportunities in collaborative business process management: Overview of recent advances and introduction to the special issue. Information Systems Frontiers. 2009 Jul 1;11(3):201-9.
12. Niehaves B, Plattfaut R. Collaborative business process management: status quo and quo vadis. Business Process Management Journal. 2011 Jun 7;17(3):384-402..
13. Camarinha-Matos LM, Afsarmanesh H. Collaborative networks: a new scientific discipline. Journal of intelligent manufacturing. 2005 Oct 1;16(4-5):439-52.
14. EFFRA. Factories of the Future 2020 Roadmap. Retrieved Apr. 8, 2016, from http://effra.eu/attachments/article/129/Factories%20of%20the%20Future%202020%20Roadmap.pdf. 2013.
15. ALICE. (2014). Information Systems for Interconnected Logistics Research and Innovation Roadmap. http://www.etp-logistics.eu/cms_file.php?fromDB=11627&forceDownload.
16. Erol O, Mansouri M, Sauser B. A framework for enterprise resilience using service oriented Architecture approach. InSystems Conference, 2009 3rd Annual IEEE 2009 Mar 23 (pp. 127-132). IEEE.
17. Antunes P, Mourão H. Resilient business process management: Framework and services. Expert Systems with Applications. 2011 Feb 28;38(2):1241-54.