An Approach for Resilient Systems Analysis

William Excoffon, Jean-Charles Fabre, Michaël Lauer

An Approach for Resilient Systems Analysis

William Excoffon, Jean-Charles Fabre[1], Michael Lauer[2]
LAAS-CNRS, Université de Toulouse, CNRS, INP[1], UPS[2], Toulouse, France
{william.excoffon, jean-charles.fabre, michael.lauer}@laas.fr

***Abstract*—Fast evolution of computing systems is still a challenge today, but it is becoming now an issue for safety critical embedded systems. The challenge here is to maintain dependability properties when facing changes. This is exactly the definition of resilient computing we consider in this work. The objective of the paper is to simulate such changes using models to measure the resilience of a system and improve it.**

I. Introduction

Many systems today are subjects to changes during their operational life. In critical computing, these changes must not interfere with dependability. This capacity to remain dependable despite changes is called resilience [1]. Resilience relies at runtime on *Fault Tolerance Mechanisms* (FTMs). They requires some assumptions (fault model, application characteristics, etc.) in order to be valid, both application characteristics and fault model assumptions can change during the life of the system.

In operation, different events can occur and modify the fault model defined in a first analysis (Electromagnetic perturbations, hardware ageing, outdated software,...). If the assumptions on the fault model are not correct any more at a given point in time then the system dependability is not guaranteed by the initial set of FTMs as discussed in [2].

The objective of this work is to define and validate models and measures to analyze resilient systems. In this paper we present a way to simulate the life of a dependable system. The model and measures defined in [3] will be used to estimate the resilience capabilities of a system and to improve its resilience using Adaptive Fault Tolerance.

The terms used in this paper will be defined briefly in section II . In section III we will show how to use simulation to evaluate adaptation policies. Section IV go further in the analysis and proposes use cases for the simulation results and section V concludes this paper

II. Resilience and Model for Dependable Systems

In this work we consider the architecture of the system being component-based as defined in [4] and [**?**]. Each component need to be dependable and therefore it must be protected by a Fault Tolerance Mechanism (FTM). In order to apply a FTM to a component some assumptions must be made as stated in the introduction.

We defined the compatibility between a component and the FTM attached to it as the ability for the FTM to accept the application characteristics of the component such as determinism, state access,...

In the same way we defined the adequation between a component and its FTM as the ability for the FTM to tolerate all types of fault made by the component.

Finally, the consistency is the property for a component and its FTM to be both compatible and in adequation, this property can be applied to the system (if every application is consistent).

A scenario is a sequence of event. Each event induces some modifications regarding the Application Characteristics (AC) and/or the fault model of a component. When an event occurs, the modification is monitored and the services proposed by the resilient framework described in [3] checks the consistency property. When needed, it can change the FTM in order to restore this property.

Here is an evolution scenario given as an example (AC change impact):

- At t_0, a given application A, a command and control application, is attached to a FTM tolerating crash faults, say Primary Back-up Replication (PBR) to save CPU usage.
- At t_1, A1 is updated. The new version A2 is deterministic but does not offers access to its internal state anymore, invalidating the PBR.
- At $t_1 + \delta t_1$, a new FTM is assigned to A2, namely a semi active replication strategy, Leader Follower Replication (LFR), A2 being deterministic and no state access is required.

III. Simulation

A policy is a set of deterministic rules which lead to choose an FTM for a component based on the informations provided by the model. As the system must be adaptive these rules must be evaluated regarding the resilience criteria. In this section we talk about the evaluation of policies and the impact of initial configuration.

A. Policy evaluation and Initial configuration

Let's assume that based on safety analysis we have a policy to choose between several FTMs for each components. The idea is to measure how good is this policy regarding the resilience of the system. For a given set of components, we generate a set of scenario. For each event of each scenario we apply the policy to choose a FTM.

Then we measure the resilience of the system thanks to two measures. The first one is a statistical approach it shows the proportion of events the system is resilient to.

$$RE(t) = \frac{N - ic(t)}{N} \tag{1}$$

where t is the total period of observation (the lenght of the scenario), N is the number of events (AC or fault model changes) during this scenario and $ic(t)$ is the number of inconsistencies observed.

The second one is defined as the Mean Time Between Inconsistency (MTBI).

$$MTBI(t) = \frac{t - \sum_{i=0}^{ic(t)} \delta t_i}{ic(t)} \quad (2)$$

where t is the total period of observation, $ic(t)$ is the number of inconsistencies observed and δt_i the amount of time during which the system is inconsistent after change event i. With this measures we can capture the fact that some FTMs takes longer to install, reconfigure or develop.

The mean value of these measures are computed for the set of scenario and can be compared to the results obtained with other policies on the same set of scenario.

With this policy and this set scenario we can measure the sensitivity to the initial configuration. The granularity of the architecture can have a significant impact on the resilience. Therefore, the simulation can help to analyse the consequences of architectural choices like aggregating some components.

B. Policies generation

The simulation can also help us to define new policies. As stated before, sometimes the solution is not unique when choosing a FTM. Therefore we propose an exhaustive analysis of all the possibilities. Let's consider an initial set of components and a scenario. Each time an event occurs we generate a new FTM configuration for each possibility. This can be represented by a tree in which the children of a node are all the possible configurations. Each level is the representation of an event.

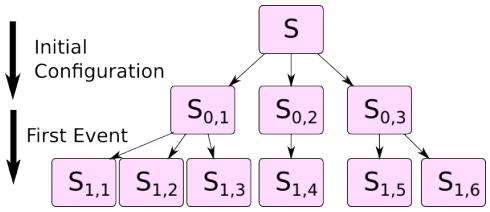

Fig. 1. System evolution for a given scenario

This tree represents all the possible choices, by measuring the resilience for each path (with the measures defined previously), we can defined some new policies and then check them on a set of scenario to be sure of their effectiveness. In order to compare two policies the costs in terms of resources must be taken into account. Otherwise, a mechanism such as Triple Modular Replication will always be better than any other FTM.

The analysis of these simulations can be used to give us more informations regarding the resilience of a system.

IV. Future applications

As the simulation is based on the generation of random events, a sensitivity analysis regarding the probability distribution function used during the generation must be conducted. The aspect of resilience can be divided between the resilience to the AC changes and the resilience to the fault model changes. The impact on the relevance of the policy chosen must be carefully evaluated.

This tool can provide us some comparison between two FTMs. Today, to choose a FTM over an other we consider the cost and the time to recover from a fault. By analysing each situation on multiple scenario where a choice must be done between two or more specific FTMs (i.e. each subtree in fig 1) we would have some informations regarding the consequences in term of resilience. The goal is to determine if a FTM is better in term of resilience and to transform this information into a rule for a policy.

Sometimes there is no solution, an analysis conducted on several scenarii could help us find what situation is a dead-end, how it happened and how we can prevent it from happening. For example, if the results shows that most of the time we need a replication protocol the component is either non-deterministic or doesn't give access to it state, the designers can prevent it by developing components either deterministic or giving acces to its state to allow a PBR or a LFR to be set.

Last but not least, as the aim is to make a more adaptive system, these simulations, using multiples scenarii, could be used as training sets for self-learning algorithms. In this case, such non-deterministic algorithms won't compromise the safety of the system because all the FTMs that could be chosen are solution to the inconsistency.

V. Conclusion

Resilient computing for autonomous systems is a hot topic today, since autonomy is spreading many critical applications (drones, automotive, robots in general) in both civil and military domains. The safe handling of the fast evolution of autonomous systems is, like for any other computer-based system, definitely mandatory. This is exactly the context of this work we carry out at the moment in close collaboration with partners in the automotive industry.

Our current work is concerned with the definition of experiments to validate the proposed model and measures, essentially based on simulations. The proposed measures will be analysed and extended. They will be used to parametrize the adaptation of FTM at runtime in a pro-active and probabilistic way.

References

[1] Jean-Claude Laprie. From dependability to resilience. In *38th IEEE/IFIP Int. Conf. On Dependable Systems and Networks*, pages G8–G9. Citeseer, 2008.

[2] David Powell. Failure mode assumptions and assumption coverage. In *Predictably Dependable Computing Systems*, pages 123–140. Springer, 1995.

[3] William Excoffon, Jean-charles Fabre, and Michaël Lauer. Towards modelling adaptive fault tolerance for resilient computing analysis. In *SAFECOMP*, 2016.

[4] Miruna Stoicescu. *Architecting Resilient Computing Systems: a Component-Based Approach*. PhD thesis, Institut National Polytechnique de Toulouse-INPT, 2013.

[5] Michael Lauer, Matthieu Amy, Jean-Charles Fabre, Matthieu Roy, William Excoffon, and Miruna Stoicescu. Engineering adaptive fault-tolerance mechanisms for resilient computing on ros. In *2016 IEEE 17th International Symposium on High Assurance Systems Engineering (HASE)*, pages 94–101. IEEE, 2016.