Folded Interaction Systems and Their Application to the Survivability Analysis of Unbounded Systems*

Michael Schiffers, Dieter Kranzlmüller
Munich Network Management Team (MNM)
Ludwig-Maximilians-Universität (LMU), Munich, Germany
michael.schiffers@ifi.lmu.de, kranzlmueller@ifi.lmu.de

Abstract. *Modeling the fulfillment of global properties like survivability is a challenging problem in unbounded systems such as Grids, peer-to-peer systems, or swarms. This paper proposes Folded Interaction Systems (FIS), an extension of the classic I-Systems framework, to overcome the modeling issues. FIS is applied to a case of survivability assessment in Grids and demonstrates the identification of essential capabilities, the modeling of harmful incidents, and the derivation of standard strategies to sustain the survival of a system's mission. FIS is not restricted to survivability, it can be used for investigating the preservation of any global property.*

Keywords. unbounded systems, survivability, interaction systems, foldings

1. Introduction

The Internet, Grids, peer-to-peer systems, and swarms over dynamic ad hoc networks are all examples of unbounded systems. They differ significantly from bounded systems as they neither exhibit a common administrative control, nor does any of their components have a complete view of the (dynamically changing) system as a whole, nor may any of its components exercise control in other system parts [1]. (We use the term "system" here in its broadest sense which not only covers hardware and software but also human resources. Examples of "components" are thus logical entities, human beings, network nodes, or complete Grid sites.)

A serious problem arises when combining systems comprising relatively isolated, small-scale elements into an unbounded conglomerate. As can be observed in Grids [2], the primary challenge is *not* the coordination of the components for joined problem solving (although difficult enough). Rather, it is the fundamental requirement to preserve the "local" system properties (e.g., security, robustness, availability) in-the-large, while at the same time to fulfill the "mission" of the system-as-a-whole defined by a set of global properties to achieve. For example, keeping an essential set of Grid services reliable does not only depend on the reliability of the underlying Grid resources but also on the relationships between them [3]. Ensuring that systems survive their *mission* – despite the presence of intrusions or disasters – is the primary objective of the discipline of *survivability* [4]. We will discuss this in more detail in section 2.

Reasoning about survivability is not possible without a formal system model that is able to express global properties, their dependencies on local interactions, the propagation of such interactions, and the transformation from non-safeness to again-safeness. Such a framework is not available today. In this paper we propose supplementing Interaction Systems (IS) [5] with foldings for closing this gap. The basic idea is to use IS for modeling systems and foldings for model transformations. *Folded Interaction Systems* will be introduced in section 3.

The suitability of the FIS modeling approach will be demonstrated in section 4 by exemplarily assessing the survivability of (an excerpt from) a production Grid authorization framework.

In section 5 we briefly compare FIS with related work before concluding the paper.

*This work has partially been funded by the Seventh Framework Program of the European Commission (Grants 246703 (DRIHMS) and 261507 (MAPPER)).

2. The Problem of Modeling Survivable Unbounded Systems

The primary objective of survivability is the system's *mission* to survive instead of single components. There are several challenges related to this (cf. [6]):

1. What is the system's mission?
2. Which essential capabilities need to survive?
3. What do they have to survive?
4. How can systems be designed with survivability already "built-in"?
5. How can system components (especially legacy components) be instrumented a posteriori to achieve survivability?
6. How can survivability be assessed in a methodologically sound manner at design time?
7. How can "survivability performance" be monitored and audited at run time?

While these challenges relate to the survivability of any system, *unbounded systems* exhibit specific constraints due to their (partial) autonomy. In unbounded systems the system components belong to different administrative domains. Typically, they are managed by component managers. These managers communicate with each other for coordinating cooperative tasks or for sharing resources, but they are independent otherwise [2]. Within their own component, however, they completely exercise control over the local processes by enabling or disabling local state transitions. The focus is hence on "distributed control" as opposed to specific functionality. The constraints summarize to:

1. Some system components may be completely autonomous (example: humans) while others may be strictly reactive (example: storage elements in Grids).
2. Every system component is aware of only a small set of other system parts (example: Grid Resource Providers only know their administrative domain).
3. System components may trigger (enforce) activities in other components (example: a Grid meta-scheduler enforces local resource managers).
4. System components may be in mutually exclusive states (examples: exclusive access to resources or forbidden inconsistencies between End Entity Certificates (EEC) and Proxy Certificates in Grids).

Related to survivability these constraints translate to questions like: Can components interact in an unintended manner? Are there unreachable global states which would partition the whole system? Is an unintended "coalition" between system components possible (which may lead to deadlocks or livelocks)? How can a system be migrated from a non-safe situation into an again-safe one?

Discussing such issues in the *same* modeling framework is not possible today as there is no such framework available. We propose Folded Interaction Systems (FIS), a combination of Interaction System (IS) [5] and structure preserving foldings to close this gap. In the next section we briefly describe IS before introducing foldings between IS-models.

3. Folded Interaction Systems (FIS)

Interaction Systems (IS) are based on the single assumption that every system component (called *part*) is in exactly one state (called *phase*) at any time. Other than related formal approaches (e.g., Petri Nets, communicating Finite State Machines, π-calculus), IS do not specify the *allowed* interactions (these would be intractable in large-scale systems). Rather, the idea is to allow everything and specify only the *restrictions* to obey. There are exactly two types of restrictions: the mutual exclusion of phases (called *coupling*), and the uni-directional *enforcement* of phases.

IS are graphically represented as depicted in Figure 1 where phases are represented by small circles and parts by rounded rectangles.

The phase a part is currently in is indicated by a filled circle (the *phase token*). The set of phases currently holding phase tokens is called a *case*. Directed edges between phases denote enforcements while undirected ones represent mutual phase exclusions. Inert parts have a gray background and they are labeled using square brackets. In Figure

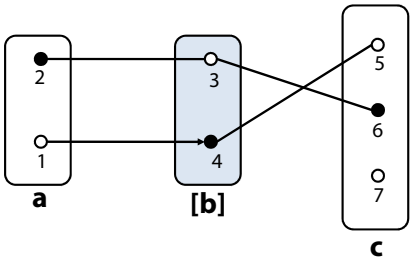

Figure 1: Example of an IS

1 (2, 3) and (4, 5) are both examples of mutual exclusion couplings, while (1, 4) expresses an enforcement from phase 1 in part a to phase 4 in part $[b]$. The semantics behind (1, 4) is that 1 exerts a force on $[b]$ to leave 4. $[b]$ will then leave 4 unless prevented by other external influences. As long as 4 has not been left, however, part a is supposed to stay in 1. Notice that phases of the *same* part are mutually exclusive by definition. More formally

Definition 1 (IS)**.** An IS is a structure $IS = (P, B, I, K, E)$ with

1. P is a finite set of phases
2. B is a partition of P
3. I is a set of inert parts ($I \subseteq B$)
4. $K \subseteq P \times P$ is a symmetric coupling relation for expressing mutual exclusion
5. $E \subseteq P \times P$ is an enforcement relation for expressing enforcements between phases ($E \cap (E^{-1} \cup K) = \emptyset$)

The dynamics of an IS is described by the "firing rule" in Algorithm 1, an axiomatic foundation of which can be found in [5].

The rule is based on a neighborhood concept. A *neighbor* of a phase $p \in b$ is any phase q in a part $b' \neq b$ which is related to p by either mutual exclusion (the coupling relation K) or by the enforcements of E and E^{-1}. A neighbor is *occupied* if it holds a phase token. For example, the phase set $\{1, 5\}$ in Figure 1 defines the neighborhood of phase $4 \in [b]$ and the occupied neighbors is empty under the given phase token distribution.

From the behavior graph in Figure 2 we derive the impossibility of a phase transition $1 \rightarrow 2$ in case $\{1, 3, 7\}$, whereas the transition $5 \rightarrow 7$ is possible. As an example of a multistage influence propagation consider a phase transition in part c from 7 to 5 in case $\{1, 4, 7\}$. The transition would induce part $[b]$ to leave phase 4 to phase 3. Please notice that part a is now unable to leave phase 1 because of the coupling between phases 2 and 3.

Algorithm 1 Local Transition Rule

Require: $p \in b$ holds current phase token
Require: p' is a potential successor

1: if neither p nor $p' \in b$ have an *occupied* neighbor then b may decide to pass the phase token on to p', provided b is autonomous (and not inert)
2: if p is enforced by an occupied neighbor phase q then b has to leave p
3: if $p' \in b$ enforces an occupied neighbor phase then b may decide to pass the phase token on to p', provided b is autonomous (and not inert)
4: if the phase token may not be passed on then all neighbors are prompted to leave their current phase
5: no other phase transitions are allowed

Enforced phase transitions often follow a local Finite State Machine (FSM) transition scheme (reflecting e.g., internal procedures or policies). We indicate this graphically by dotted arrows between phases (as in part VO in Figure 3).

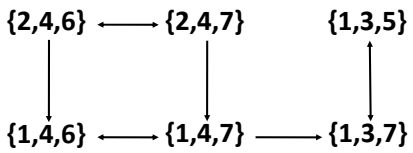

Figure 2: Behavior graph of the IS in Fig. 1

After this preparation we can now define IS-foldings as structure preserving mappings:

Definition 2 (IS-folding)**.** Let IS_1 and IS_2 be an IS with $IS_1 = (P_1, B_1, I_1, K_1, E_1)$ and $IS_2 = (P_2, B_2, I_2, K_2, E_2)$. A mapping α: $P_1 \rightarrow P_2$ is called *IS-folding of* IS_1 *into* IS_2 if it preserves the coupling and enforcement relations of IS_1 in IS_2 (i.e, $\alpha(P_1) \subseteq P_2$ and $(\alpha(p_1), \alpha(p_2)) \in K_2$ for $(p_1, p_2) \in K_1$). α is called *part respecting* if it respects B_1 and *case respecting* if IS_2 exhibits the same case transition semantics as IS_1. An IS-folding

which is both part and case respecting is called *strong*.

Folded Interaction Systems (FIS) are IS with an associated family of IS-foldings.

An example of an iterated application of IS-foldings will be given in the next section when applying FIS to the survivability analysis of a Grid authorization framework.

4. Applying FIS to Analyze the Survivability of a Grid Authorization Framework

Loosely, Grid authorization is the act of providing and checking the authority of a user or a Grid job on a specific set of Grid resources. An IS model ($DGAF_0$) of (an excerpt from) the D-Grid authorization framework ($DGAF$) [7] is shown in Figure 3.

Resources (the inert part [$Resource$]) are made available to Virtual Organizations (VOs) (part VO) by Resource Providers. Access to resources is granted to VO members according to their "position" relative to the VO. This position is defined by group memberships (inert part [$Group$]) and the role tenancies (inert part [$Role$]) within each group. Positions are managed and published by the Virtual Organization Membership Service (VOMS) (inert part [$VOMS$]). A registration in VOMS implies a registration for a default role in a default group. The attributes that unambiguously identify VO members (and thus implicitly Grid jobs executing on behalf of them) are encoded in X.509 End Entity Certificates (EEC) (inert part [EEC]) and proxy certificates (inert part [$Proxy$]) derived from EECs. Proxies are signed by an attribute authority, in this case VOMS. VO memberships have a life cycle which is expressed by the FSM-driven behavior in part VO. Finally, a (middleware specific) Grid Job Manager allows only valid Grid jobs (part [$Job\ Mgr$]) to "consume" Grid resources.

Starting from the initial global state (indicated by the black phase tokens in $DGAF_0$) we can easily derive several global properties of $DGAF$.

1. Resource access is only granted upon presenting a valid proxy certificate (based on a valid EEC) and the corresponding EEC owner has to be registered in VOMS as a VO member (an "applicant" status is not enough).
2. Resource access may nonetheless be denied even for a valid Grid job. A typical scenario would be the unavailability of a required resource.
3. VO applicants will get member status once they are registered in the VOMS system.
4. VO memberships need to be deleted from the VOMS system upon membership termination (enforcement $(member, registered)$).
5. Any invalidation of an EEC requires the immediate co-invalidation of all derived proxy certificates.

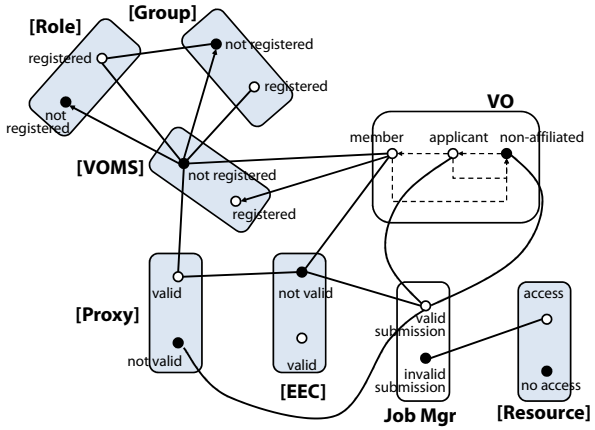

Figure 3: Excerpt from the D-Grid authorization framework [7] (Model $DGAF_0$)

From the discussion before it should be obvious that further restrictions (e.g., certificate revocation lists, credential repositories) can be added *incrementally*.

Survivability assessment generally follows a multistage process [6] consisting of an essential property assessment, an incident assessment, and a strategy definition to sustain survivability. In the following we will briefly demonstrate how FIS can be applied to support these stages.

Essential Property Assessment

One (there may be many) $DGAF$ mission statement is `"to avoid resource access grants without a valid certificate"`.

This mission neither requires VOMS to execute nor a VO to operate. It only requires the mutual exclusion of the phases $not\ valid \in [EEC]$ and $access \in [Resource]$. In FIS terms this translates into the specification of a strong IS-folding the target IS of which ($DGAF_1$) is given in Figure 4.

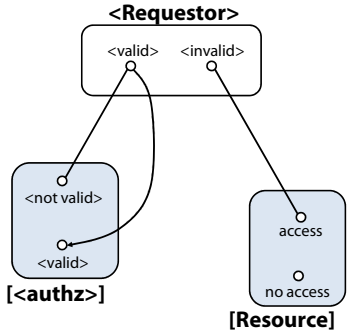

Figure 4: Essential property for $DGAF$ derived from IS-foldings (Model $DGAF_1$)

The folded parts and phases are indicated by angle brackets and are renamed appropriately. Without going into details, $DGAF_1$ was derived from $DGAF_0$ by folding the inert parts $[Role]$, $[Group]$, $[VOMS]$, $[EEC]$ and $[Proxy]$ into $[< authz >]$ and the autonomous parts VO and $Job\ Mgr$ into $< Requestor >$. Please note that the $DGAF$ mission still holds in $DGAF_1$.

Incident Assessment

In FIS we are able to express incidents as deviations from the *nominal* model defined at design time to specify the "correct service" [4]. Deviations either occur as illegal *restrictions* or as invalid *extensions*.

For example, an unintended permanent resource access can be *forced* – provided this was granted before once – by adding a part construct like the one in Figure 5 (part "$Disaster''$"). In a similar way, accessing a Grid resource despite an invalid EEC can be achieved by adding one or more uncoupled phases to part $Job\ Mgr$ allowing for unintended transitions. Generally, incidents are (in FIS terms) "combinations " of IS (the nominal one and models describing intrusions) integrated by IS-foldings.

It should be noticed that the detection of incidents requires suitable mechanisms to distinguish "invalid" behavior from intended behavior (*self/non-self tolerance*). This topic will be addressed in more detail in a follow-up paper.

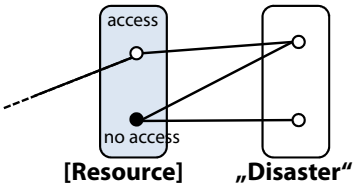

Figure 5: Unintended permanent resource access (behavior restriction)

Strategies for Sustaining Survivability

Formally speaking, mission fulfillment means transforming the system from a non-safe global state into an again-safe global state with at least all essential properties preserved. There are several strategies to achieve this transformation. They can be derived directly from the model. We briefly mention just three.

1. The *cut-off* strategy aims at achieving mission survivability by cutting off system parts. It assumes undamaged essential properties .

2. The *peripheral tolerance* strategy aims at extending the system in such a way that the damaging parts (the periphery) are tolerated.

3. The *degeneracy* strategy aims at "multiply providing" some (or all) critical system components over structurally *different* parts (degeneracy).

An in-depth discussion of transformation strategies is beyond the scope of this paper and will be presented elsewhere.

5. Related Work

The FIS-approach is based on the IS framework which has been shown to be more expressive than comparable modeling methodologies like Petri Nets or communicating FSM [5]. The latter ones suffer from inherent difficulties in enunciating restrictions, violations, and enforcements. Additionally,

the intrinsic possibility of incrementally modeling in-the-large reveals a further advantage of IS. The IS framework has, however, not been applied for modeling property preservation (like survivability) in "compromised environments". The folding mechanisms we presented here are examples to close this gap.

Survivability of IT systems, on the other hand, is a relatively new research area with a precise definition of what to achieve and a common understanding of the means how to achieve the goals still lacking. Nonetheless, there are already some more or less mature architectures available with mostly domain specific modeling frameworks [8]. None of them, however, provides a formally sound modeling framework or a practical methodology for dealing with incremental restrictions. Emergent algorithms have been proposed in [1] for achieving survivability. Although a promising approach, they require a global observer and do not provide an adequate model for reasoning about propagation of influences.

Related work is also performed in the autonomic computing community when studying self*-mechanisms [9]. The autonomic models include very interesting control loops constructs but do not allow reasoning about mutual exclusions of local states and influence propagations.

6. Conclusion and Further Work

Survivability is a global system property. A problem arises when studying survivability in unbounded systems as there is no central control and the system components only have a limited view on the system-as-a-whole. In such systems global properties can only be achieved by purposefully influencing bilateral interactions to propagate. A comprehensive modeling framework for investigating the global effects of local activities is missing. We proposed Folded Interaction Systems (FIS) as such a framework. FIS uses the Interaction Systems framework for describing system structures and interactions between components and introduces the concept IS-foldings to derive essential system capabilities, to describe incidents, and to derive survivability sustaining strategies. We demonstrated the appropriateness of FIS by applying it to the analysis of a production Grid authorization framework.

The work presented here is a first cornerstone of a comprehensive "survivability toolkit" for unbounded systems including runtime incident detection capabilities and dynamic overwrite mechanisms.

References

[1] D. A. Fisher, H. F. Lipson: *Emergent Algorithms: A New Method for Enhancing Survivability in Unbounded Systems.* Proceedings of HICSS '99, 1999.

[2] I. Foster, C. Kesselman, S. Tuecke: *The Anatomy of the Grid: Enabling Scalable Virtual Organizations.* International Journal of High Performance Computing Applications. Vol 20, No 3, 2001.

[3] C. Dabrowski: *Reliability in Grid Computing Systems*, Concurrency and Computation: Practice and Experience 21(8), 2009

[4] A.Avizienis, J. Laprie, B. Randell: *Fundamental Concepts of Dependability.* Technical Report 01-145, LAAS-CNRS, 2001.

[5] H. F. Wedde, A. Wedig, A. Lazarescu, R. Paaschen, E. Rotaru: *Modeling and Analyzing Large-Scale Distributed Interaction.* Technical Report 793, Technical University Dortmund, Germany, 2004.

[6] J. C. Knight, E. A. Strunk, K. J. Sullivan: *Towards a Rigorous Definition of Survivability.* Proceedings of the DARPA Information Survivability Conference and Exposition, Washington DC, 2003.

[7] P. Gietz, C. Grimm, R. Gröper, S. Makedanz, H. Pfeiffenberger, M. Schiffers, W. Ziegler: *A Concept for Attribute-Based Authorization on D-Grid Resources*, FGCS 25(3), 2009.

[8] P. Tarvainen: *Survey of the Survivability of IT Systems*, Proceedings 9th Nordic Workshop on Secure IT-Systems, Helsinki/Finland, 2004.

[9] M. C. Huebscher, J. A. McCann: *A Survey of Autonomic Computing – Degrees, Models, and Applications*, ACM Computing Surveys 40 (3), 2008