Spider Diagrams for Policy Modelling

P. Bottoni / A. Fish / A. Heußner / F. Parisi Presicce

General Goal

In the context of safety and security of dynamic systems, policies propose a natural way to frame a system's behaviour.

Here, we propose a formal modelling framework for specifying policies and reasoning about policy conformance. We aim at both:
(i) a rigorous formal underpinning for reasoning and
(ii) an "intuitive" visual front end for the domain expert.

Parking Domain

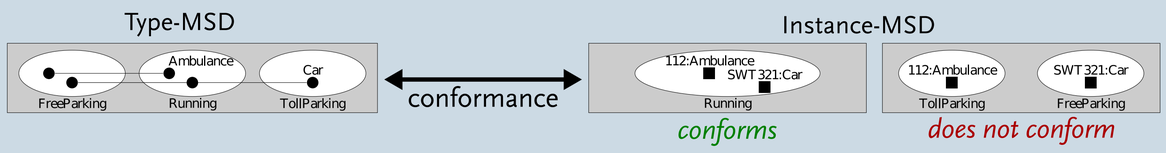
Parking places: P (toll parking), P (toll)free parking. Vehicle states: running, toll parking, free parking.

Example policy rules:
- Ambulances cannot be in toll parking.
- If a Car is in free parking, it is occupying a free parking space.
- A Car can stay at most 60 min in free parking, and before returning, it must be either running or stay in toll parking for at least 120 min.

Timed MSDs

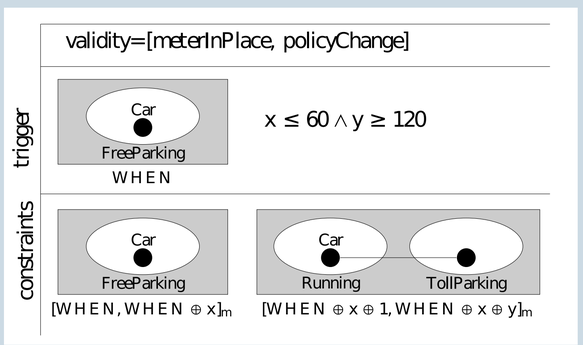
Goal: Express policies with temporal constraints.

Step 1: Adapting Spider Diagrams to OO Modelling leads to two classes of Modelling Spider Diagrams (MSDs)



Step 2: Embed notion of Timed MS in policy framework

A *Policy* is a triple
- validity
- trigger
- set of constraints



Annotating spiders with expressions over interval algebra permits the definition of permissible states of objects in time.

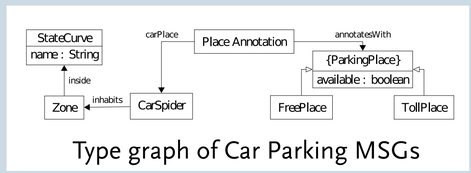
Results: Full featured formalism to specify and reason on policies based on novel Modelling Spider Diagrams enhanced with interval annotations.

References:
P. Bottoni, A. Fish. "Extending Spider Diagrams for Policy Modelling". J. Vis. Lang. Comput., 24(3), pp169–191, 2013

Policies with Resources

Goal: Express dependency of evolution of elements under a policy from additional resources (e.g., *right type of parking place available*).

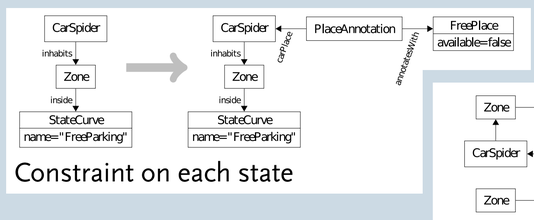
Step 1: From MSDs to Modelling Spider Graphs (MSGs)



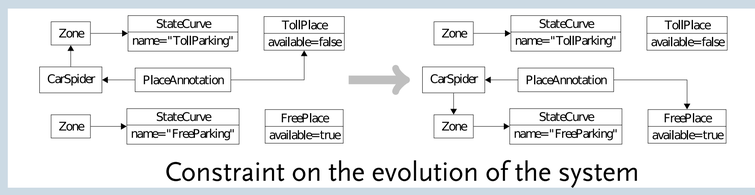
Type graph of Car Parking MSGs

Directly express relation to resources in MSGs by annotations.

Step 2: Synchronization of evolution via grammar rules



Constraint on each state



Constraint on the evolution of the system

Results: Proper foundation of policy reasoning (including resource annotations) via graph-transformation rules.

Extends:
P. Bottoni, A. Fish, F. Parisi Presicce. "Spider Graphs: A Graph Transformation System for Spider diagrams", SoSySm, 2014 (to appear)

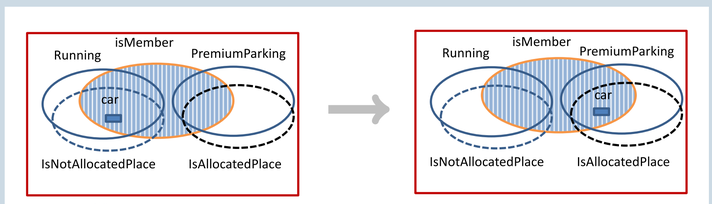
Methodology

Combine the visual advantages of (extensions of) Spider Diagrams (SDs) adapted for modelling purposes with the reasoning power of graph transformation systems.

Coloured (M)SDs

Goal: Distinguish different *types* of annotations in MSD via their colour to aid the policy modeller.

Idea: Represent different orthogonal annotation layers with the help of colouring.



Complex constraint on the evolution of the system

Annotation axes:
- Car states (Running, PremiumParking)
- Parking place states (allocated or not)
- Being a member (e.g., having a membership card)
- Length of both car and parking place

However, readability depends on additional conventions to be developed.

Results: Enlarged visual depth for MSDs.

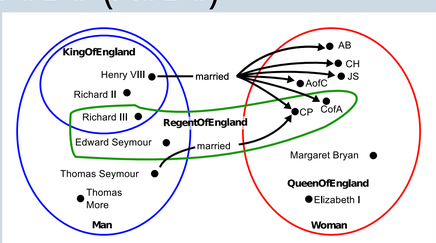
References:
P. Bottoni, A. Fish, A. Heußner."Coloured Modelling Spider Diagrams". Proc. of DIAGRAMS 2014.

Annotated MSDs

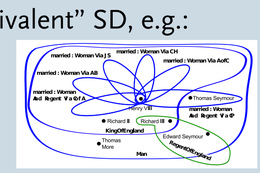
Goal: Express annotations, e.g., the relation of domain elements to resource elements, directly in MSDs while remaining in a subset of SDs that can be handled by existing SD-reasoning techniques.

Step 1: Introduce Annotated SDs (ASDs)

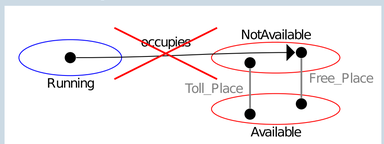
Enhance SD with annotation "arrows" that define directed relations between elements of a hiearchy of domains.



Every ASD can be translated to an "equivalent" SD, e.g.:

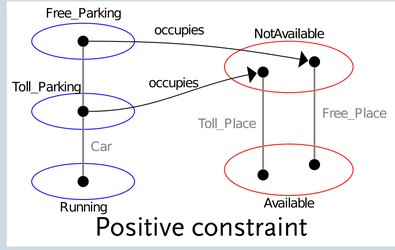


Step 2: Express constraints with Modelling ASDs



Negative constraint

Directly represent negative application conditions in diagrammatical setting.



Positive constraint

Results: Lift foundation in graph transformation rules back to Spider Diagrams via novel notion of Annotated (Modelling) Spider Diagrams.

References:
P. Bottoni, A. Fish, A. Heußner. "Annotating Spiders with Resource Information". Proc. of VL/HCC 2014.

Ongoing

- Different modes of interactions of different policies.
- Proposing generic policy specification and reasoning framework.
- Specification of constraints by adding temporal logic operators.
- Tool support.

Collaboration

Principal researchers are Paolo Bottoni (Sapienza University of Rome) and Andrew Fish (University of Brighton) joined by Alexander Heußner (University of Bamberg) and Franceso Parisi Presicce (also Sapienza University of Rome).

Sapienza Università di Roma
University of Bamberg
University of Brighton, School of Computing, Engineering and Mathematics