CERN

Outline

1. **Availability Modelling**
2. **Why Availsim?**
3. **Availsim History**
4. **Availsim3**
    1. **Terminology**
    2. **Idea**
    3. **Performance**
5. **Modeling example**
6. **Repository + outlook**

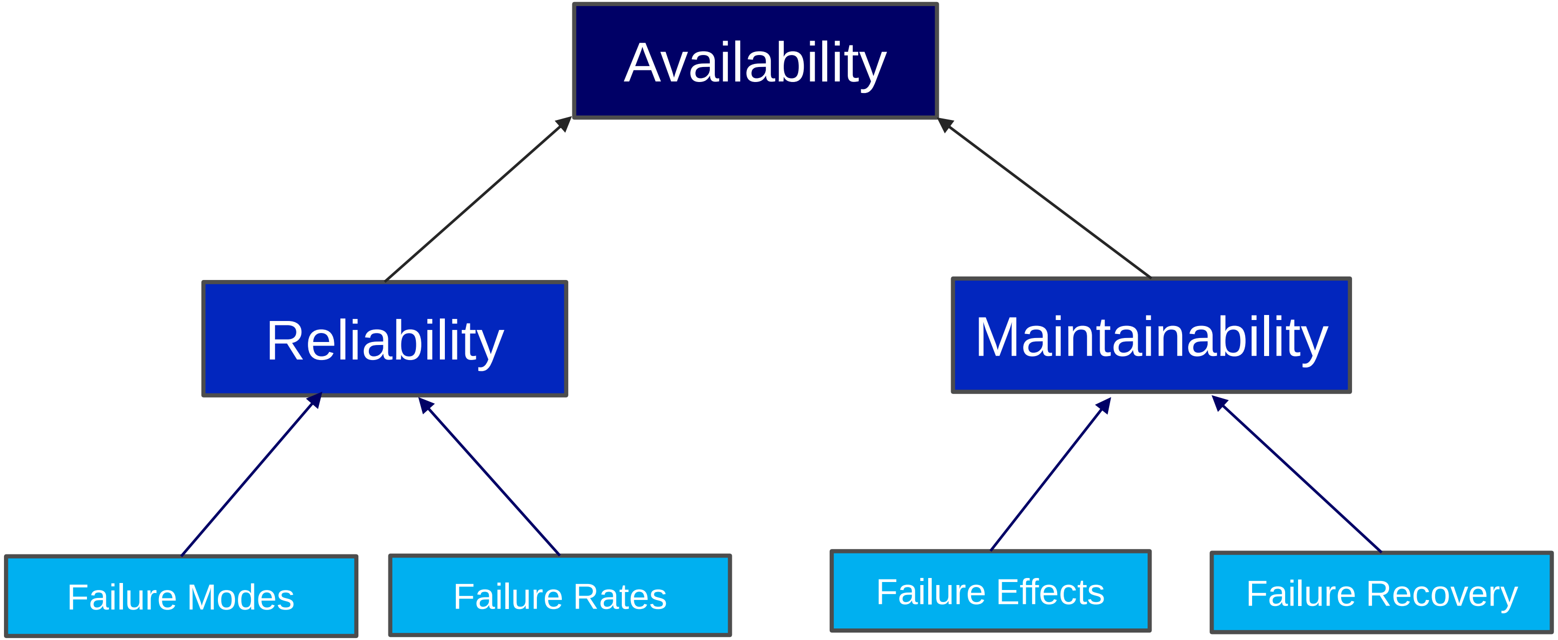
Motivation: Availability Modelling

- Availability is one of the *key performance indicators* for present and future CERN machines (LHC, Linac4, HL-LHC, CLIC, FCC)

- We need a way to predict the expected availability of these machines

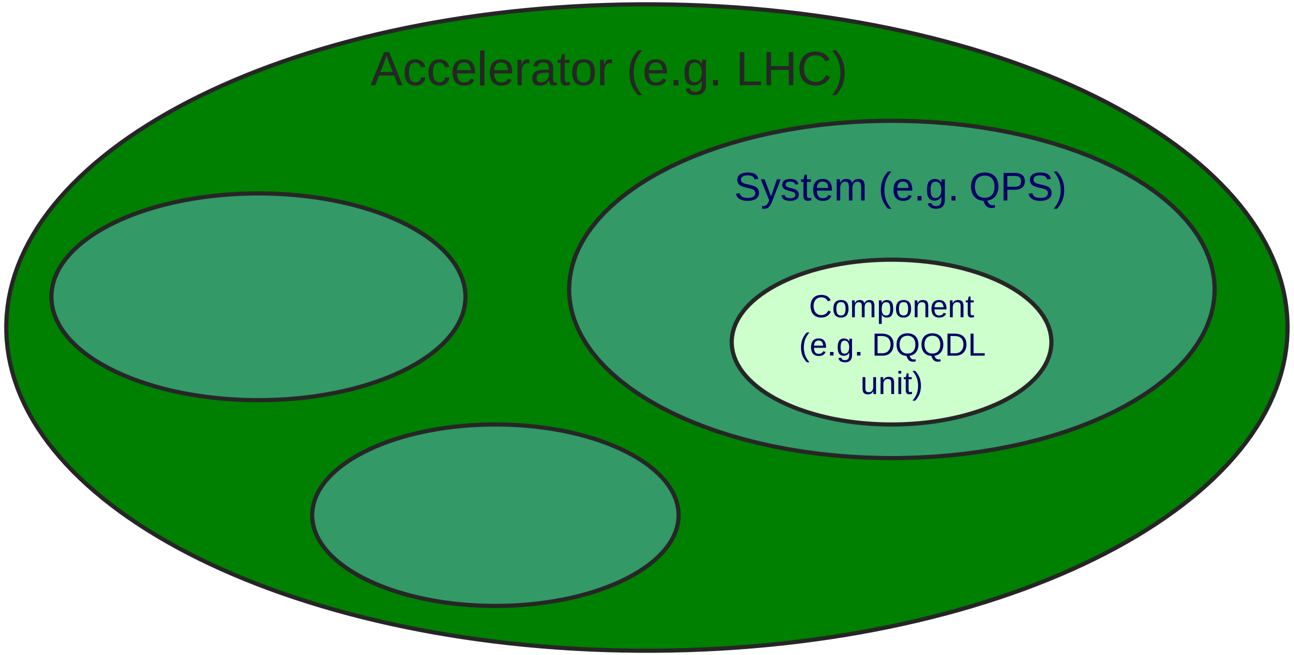
- Very different machines (linear vs circular, hadron vs lepton) + wide range of operating conditions to be covered

Basic Terminology



Failure behaviour and recoveries are described by stochastic processes (probability density functions)

Abstraction Levels



- Availability models typically start from accelerator and system levels (repairable systems) and can go down to component level

Comparison of Different Available Tools

	AvailSim	Isograph	ELMAS	Matlab*
Accelerator-specific features				
Interface for input data	Common Input Format	Common Input Format**	OpenMars	N. A.
Performance				
Licensing				
Model Definition	Common Input Format		OpenMars	N. A.
Modularity	? To be assessed	N. A.	N. A.	
Maintainability	? To be assessed			

*or any programming language that could serve the purpose

** structure can be imported in Isograph for reliability block diagrams

Availsim History

Availsim

[Why was it created?]

Commercial software

- Mainly based on RBD
- Usually productivity oriented
- Some accelerator specific needs are not considered/could not be modeled easily

→

Special-made software

– Availsim (the only software found at the time that explicitly target particle accelerators)

Availsim 1 & 2

Availsim 1

- Tom Himel and his team
- SLAC, written in MatLab
- Includes complexities not possible to model in commercial tools

Availsim 2

- Tailor-made for IFMIF RAMI analysis
- Also in MatLab
- Added additional functionalities
- Very specific for IFMIF
- Hard to manage/understand (even with in-depth knowledge)
- Not well written
- Require MatLab license

Availsim3

- Developed from scratch in Python 3 (no MatLab license)
- Object oriented
- Open sourced
- A simulation in discrete time that uses a so called ”three-phased” approach (Pidd, 1998).
    - Discrete Event Simulation (DES)
- Under continuous development and testing.
- Adding accelerator driven features

Events

● = Timed Events

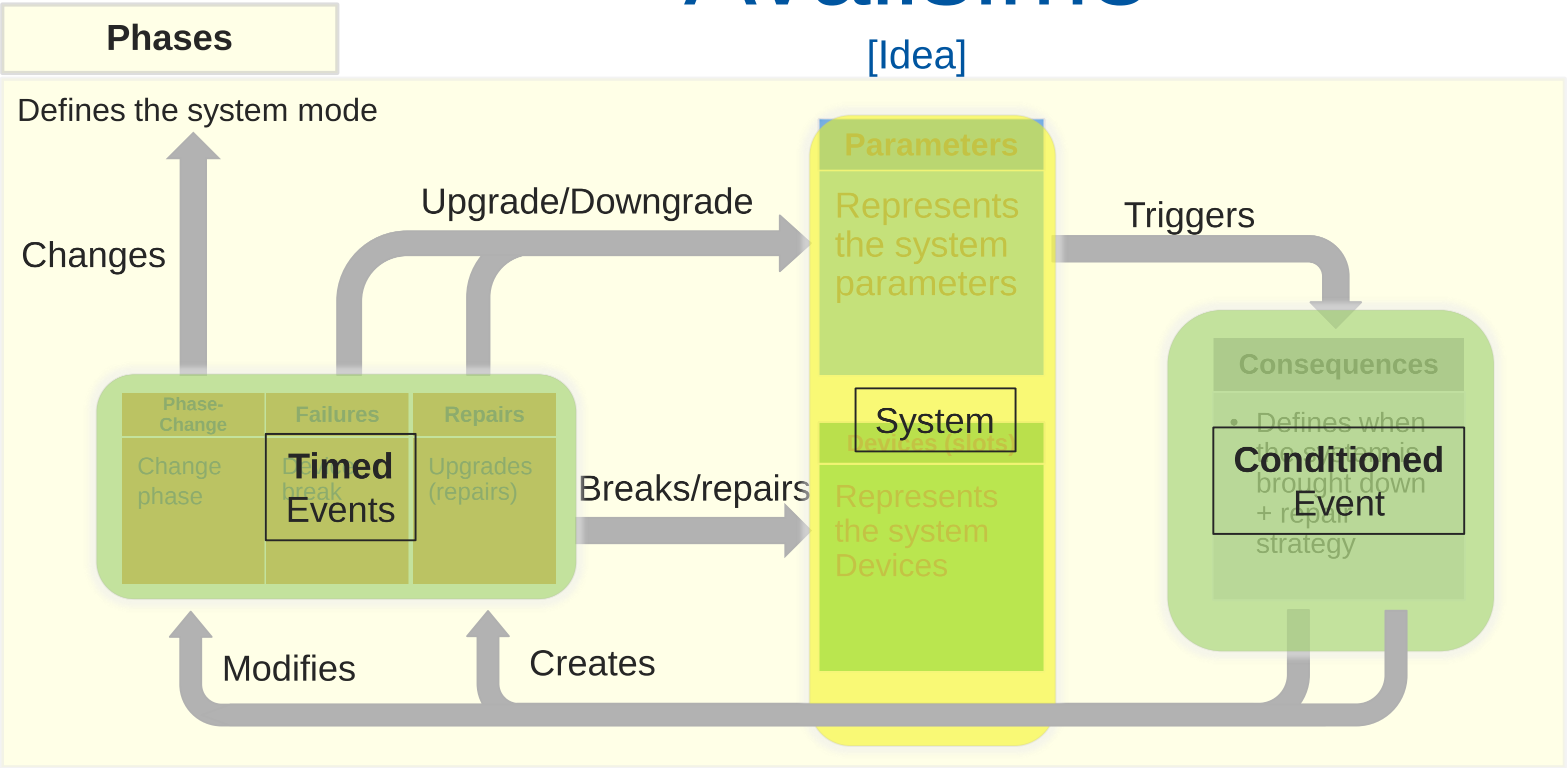
● = Conditioned Events

- Failure Events → Represents a failure in the system.
  E.g a Klystron failure, affecting beam parameters
- Consequence Event → Represents when the machine is brought down for repairs.
  e.g. klystron replaced with a new module
- Upgrade Event → Represents the moment in time
  where a repair is completed.
  e.g. power can be delivered to the cavity after klystron replacement
- Phase-Change Event → Represents the moment in time where
  the phase of the system is changed.
  e.g. Operation phase → Maintenance Phase

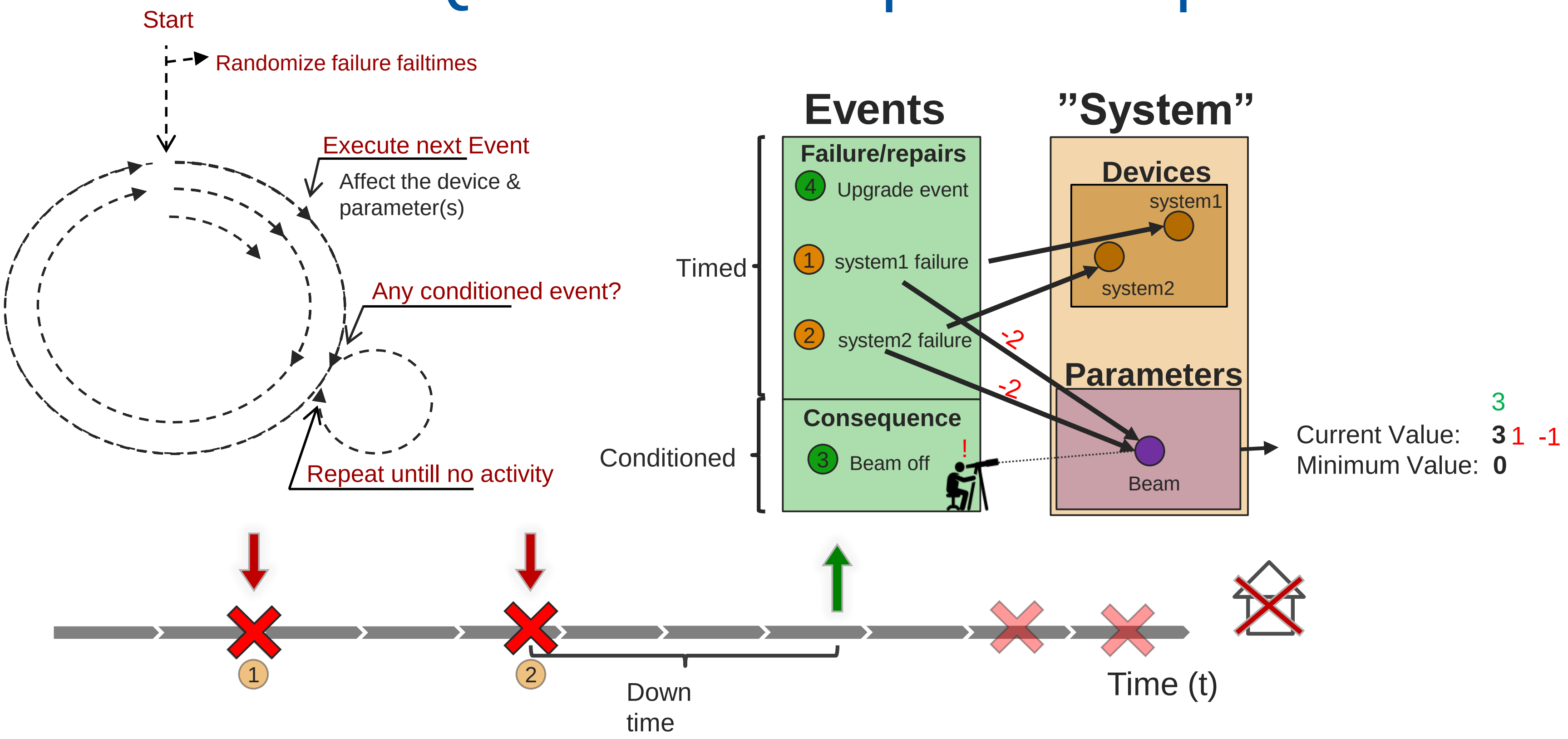
  → The phase change can also occur on condition, when
  Modified by a consequence.
  e.g. beam dump leading to injection of new beams

Availsim3

[Idea]

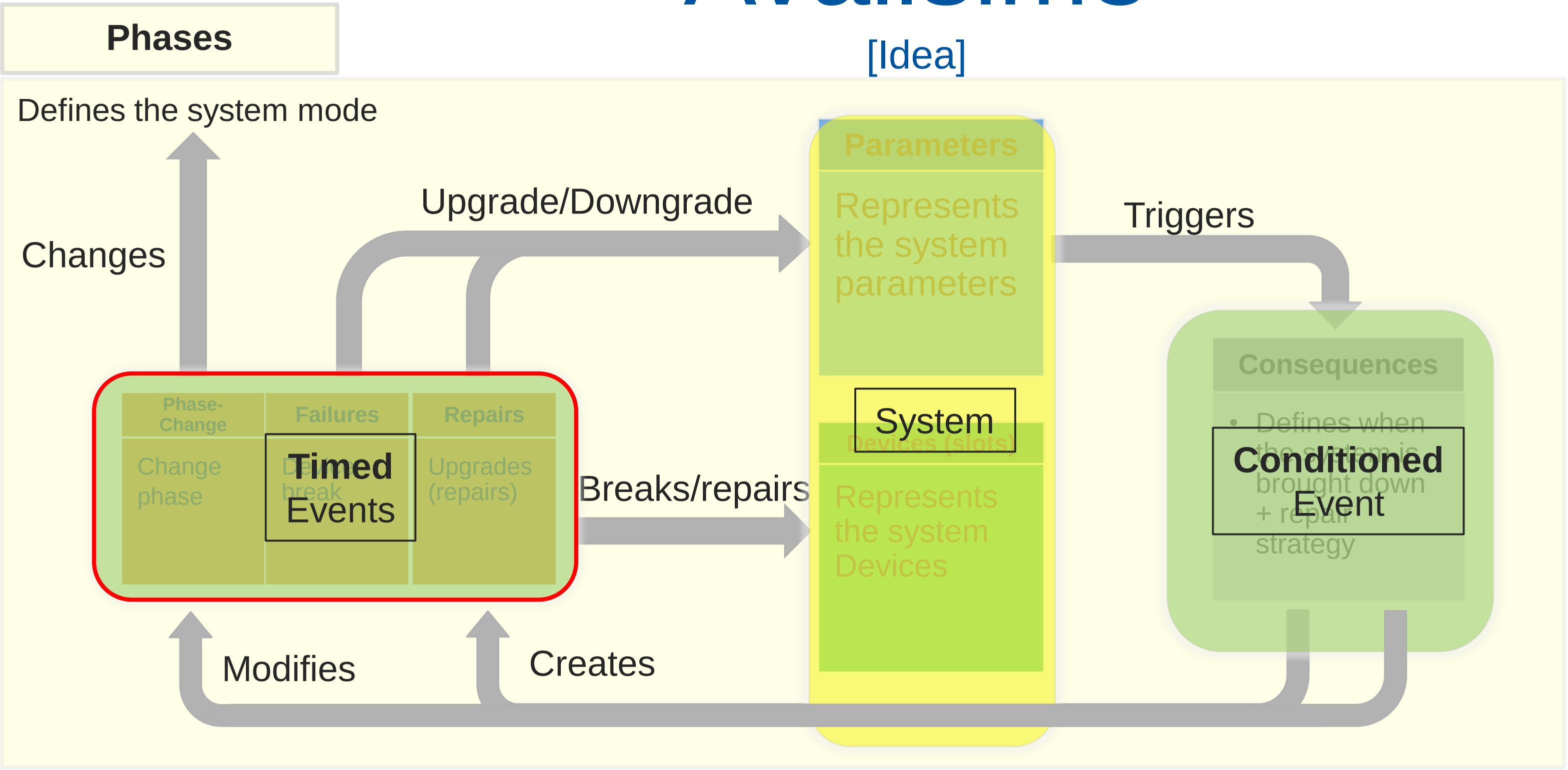


Quick and simple example

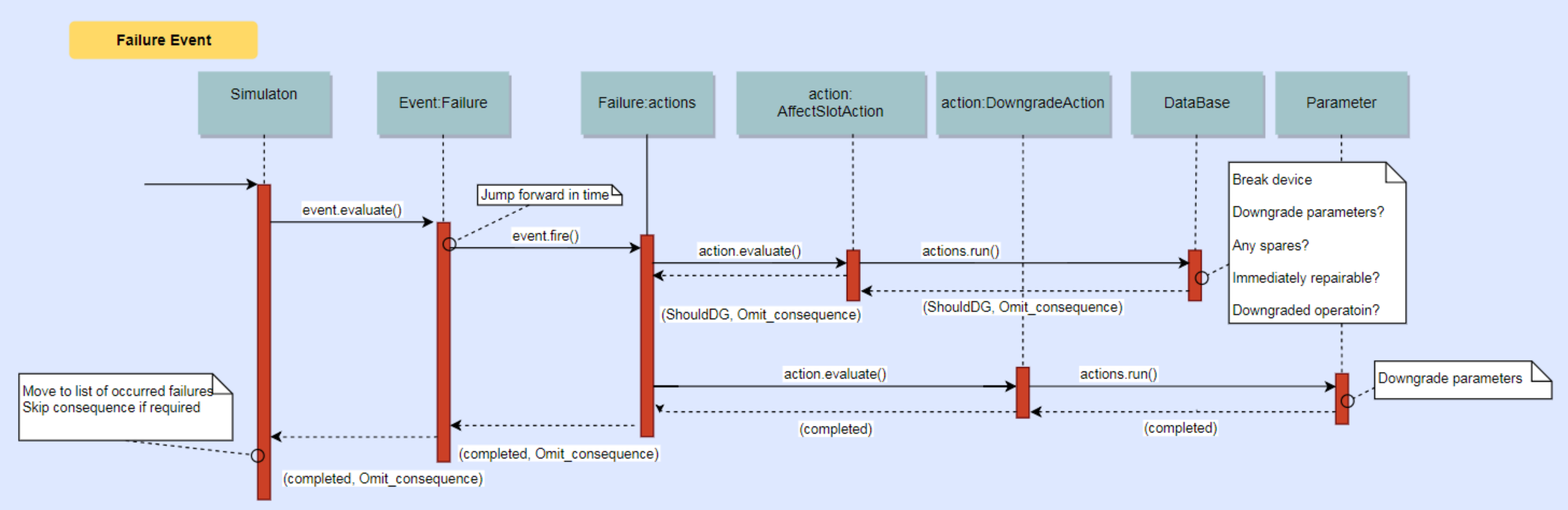


Availsim3

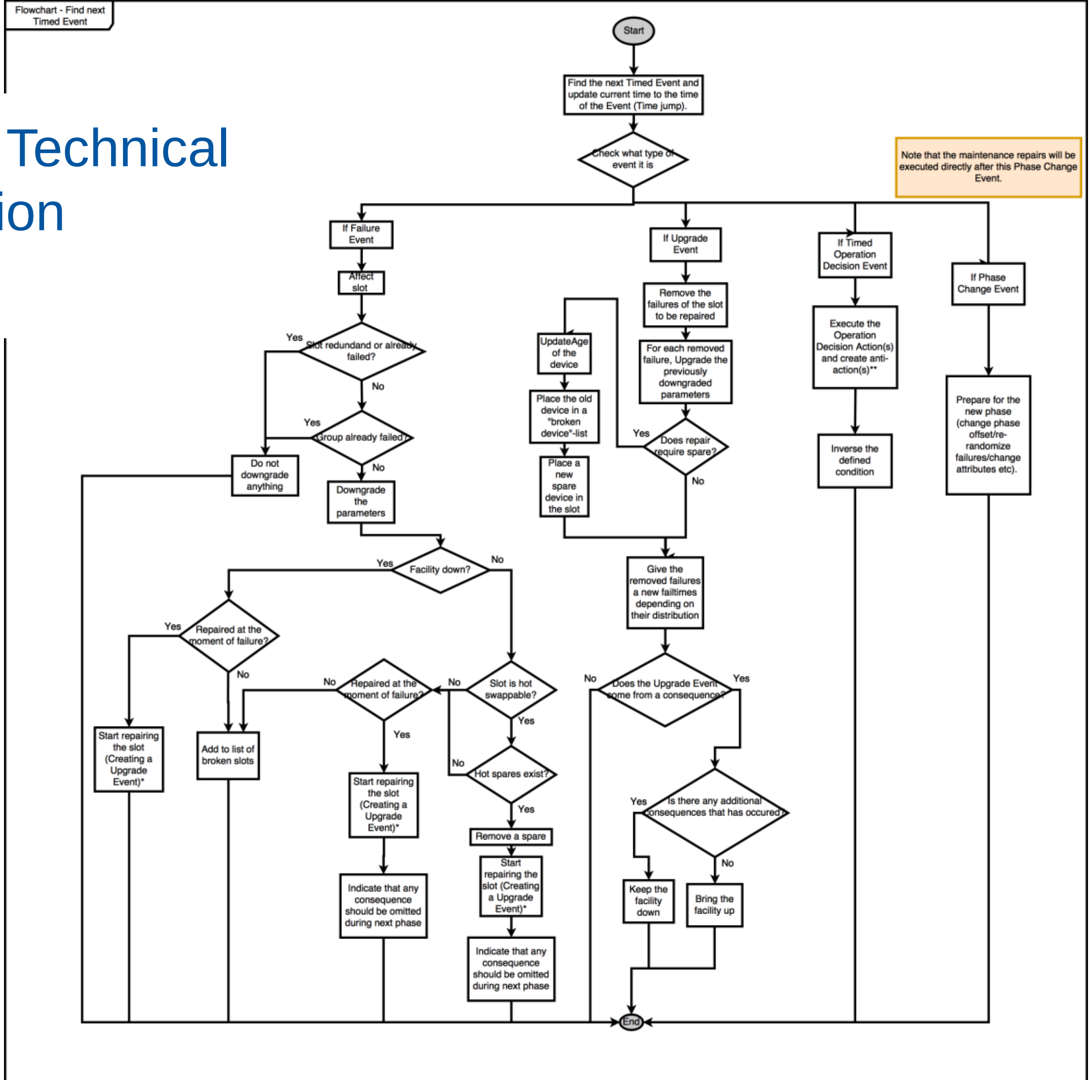
[Idea]



Failure Event – Technical Description
[1/2]

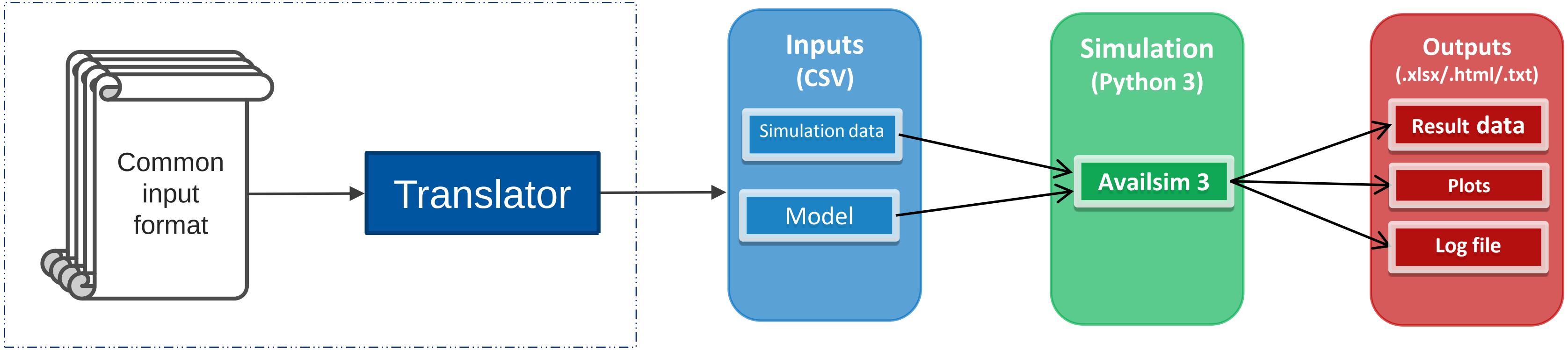


Failure Event – Technical Description
[2/2]



Availsim3

[Overview]



Modeling example - CIF

The CLIC Drive Beam Quadrupoles powering System

48 sector (24 sector / linac)

860 Quadrupole magnets / Sector = 41280 QD

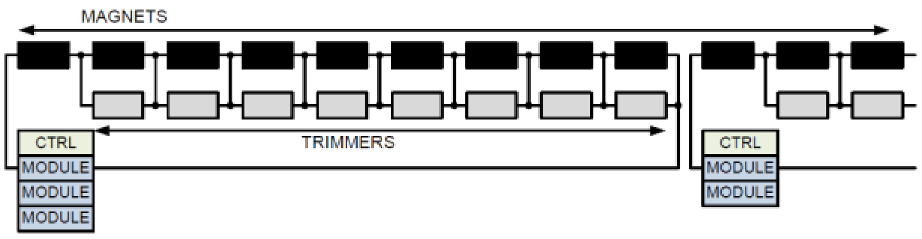
Magnet strings powered in series using trimmers to decrease current

830 trimmers / Sector – 20 failure tolerance

100 power converter modules / sector
- 12 PC of 4 modules
- 16 PC of 3 modules
- 2 PC of 2 modules

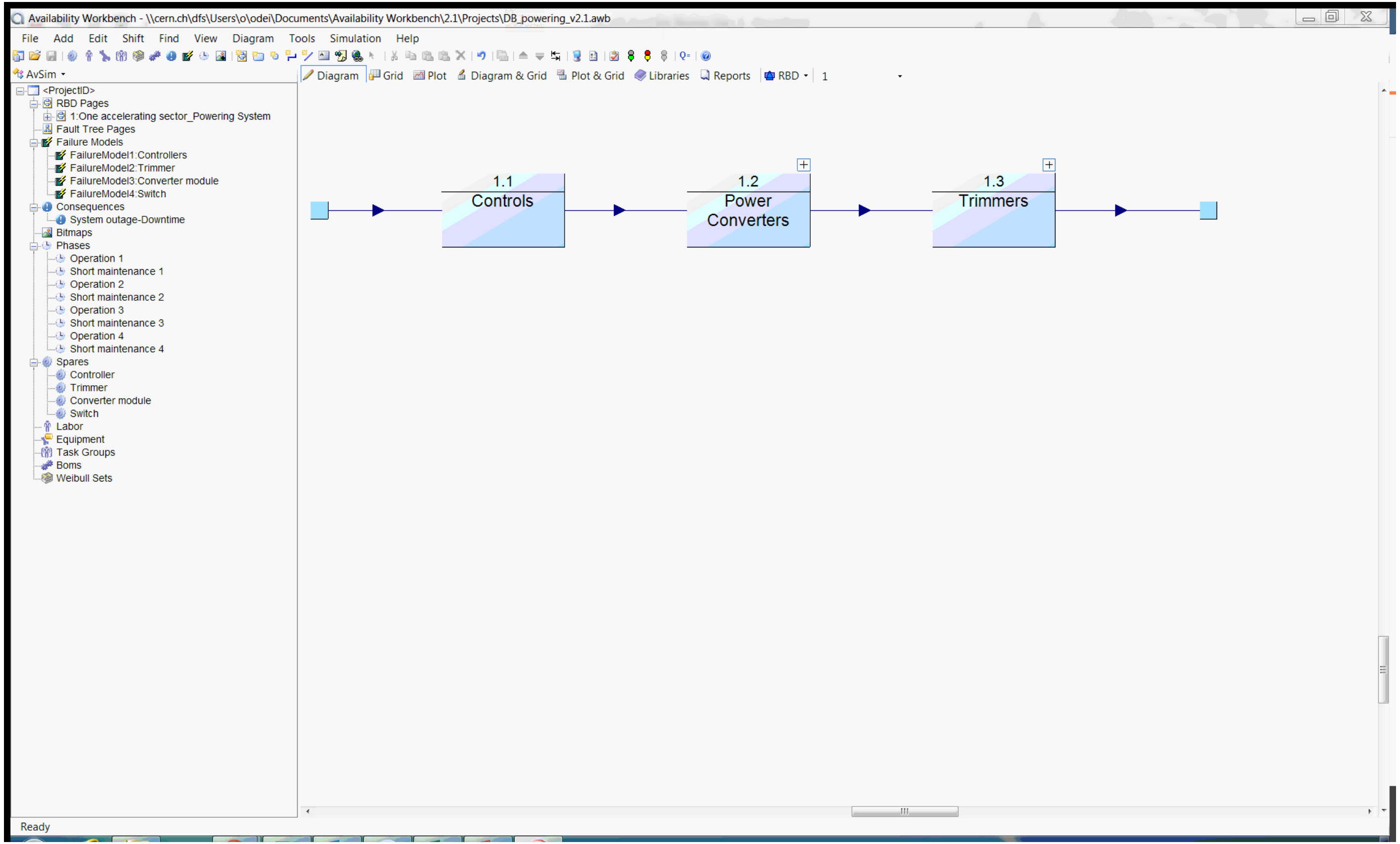
N+1 redundancy in Power converters modules

1 controller per PC (30 /sector)



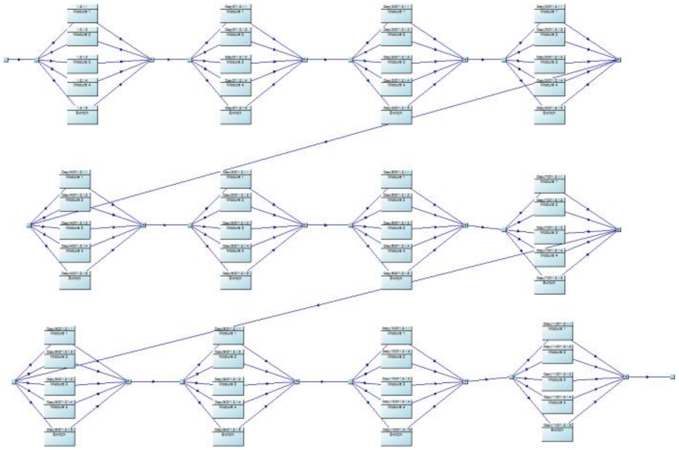
The CLIC Drive Beam Quad powering System: Availability model in **Isograph**

Only one accelerating sector!

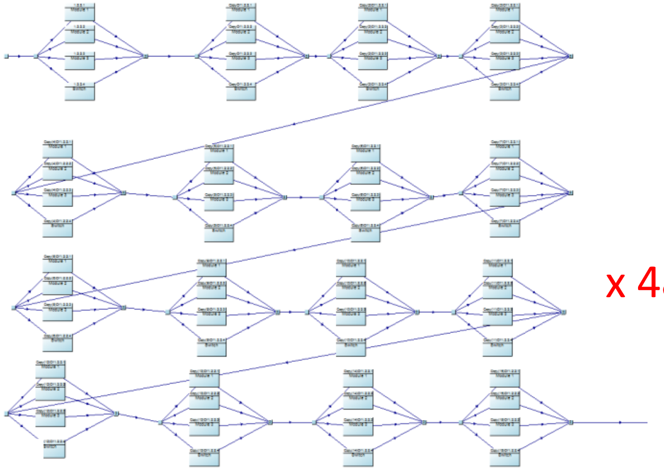


x 48

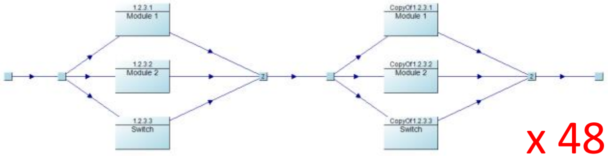
The CLIC Drive Beam Quad powering System: Availability model in Isograph



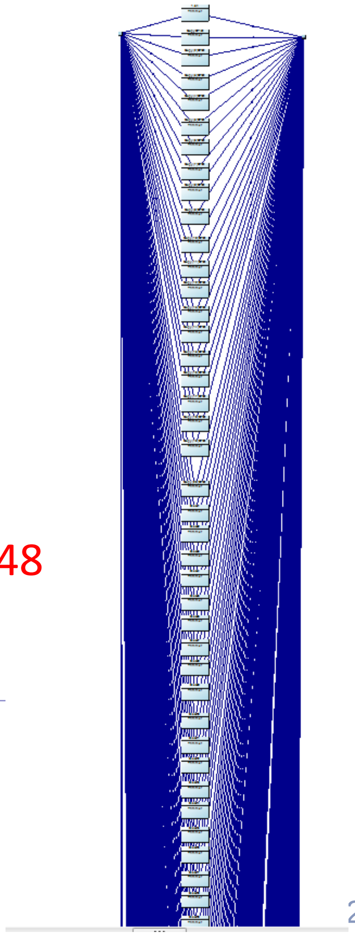
x 48



x 48



x 48



x 48

The Common Input Format for availability modelling

User
Expert

System Hardware description

Failure Modes

Failure mode Assignments

Simulation

Systems

Spares

Phases

Phases Transition

Machine Parameters

Manpower

Consequences

Locations

Modelling the CLIC DB Quads powering in the Common Input Format

System hardware and failure logic definition (positive logic)

Facility	Element	Name	Parent	Component Code	Instances	Location	Impacted System	DeviceType	Failure Behaviour Logic
CLIC	Compound	**DB QD powering**	ROOT	DB-QD	1				AND
CLIC	Compound	**Sector**	**DB QD powering**	Sector	48				AND
CLIC	Basic	**Controls**	**Sector**	Controls	1	Tunnel	Controls	Controls	
CLIC	Compound	**Power Converters**	**Sector**	PC	1				AND
CLIC	**Compound**	**4-modules PC**	**Power Converters**	4mod	12				AND
CLIC	Compound	4module	**4-modules PC**	4mod	1				AND
CLIC	Basic	Module	4module	M	4	Tunnel	Power Converters	PC Module	
CLIC	Basic	Controls	**4-modules PC**	4mod	1	Tunnel	Power Converters	Controls	
CLIC	**Compound**	**3-modules PC**	**Power Converters**	3mod	16				AND
CLIC	Compound	3module	**3-modules PC**	3mod	1				AND
CLIC	Basic	Module	3module	M	3	Tunnel	Power Converters	PC Module	
CLIC	Basic	Controls	**3-modules PC**	3mod	1	Tunnel	Power Converters	Controls	
CLIC	**Compound**	**2-modules PC**	**Power Converters**	2mod	2				AND
CLIC	Compound	2module	**2-modules PC**	2mod	1				AND
CLIC	Basic	Module	2module	M	2	Tunnel	Power Converters	PC Module	
CLIC	Basic	Controls	**2-modules PC**	2mod	1	Tunnel	Power Converters	Controls	
CLIC	Compound	**Trimmers**	**Sector**	T	1				ACTIVE(810,830)
CLIC	Basic	Trimmer	Trimmers	Trimmer	830	Tunnel	Power Converters	Trimmer	

Modelling the CLIC DB Quads powering in the Common Input Format

Failure modes definition

Failure Mode Name	Distribution	Parameters	StandbyState	Corrective Maintenance MTTR	On-Off Site maintenance	RepairStrategy	Reference System	Simultaneous repairs? (y/n)
Controls failure	exponential	3000000	COLD	4.00	on	Repairable	1	y
Trimmer failure	exponential	300000	COLD	4.00	on	Repairable	1	y
Converter module failure	exponential	300000	COLD	4.00	on	Repairable	1	y

Failure mode assignments

Component Name	Failure mode	(Operational)Phase
Controls	Controls failure	OP
Trimmer	Trimmer failure	OP
Modules	Converter module failure	OP

Simulation parameters

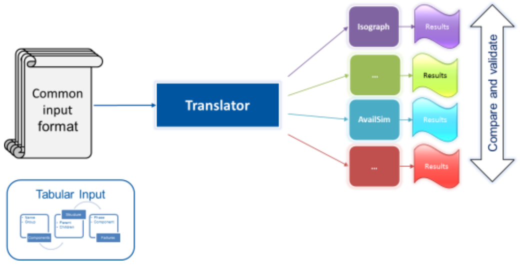
Facility	SimHours	No.ofSimulations	Seed
CLIC	2400	100	10

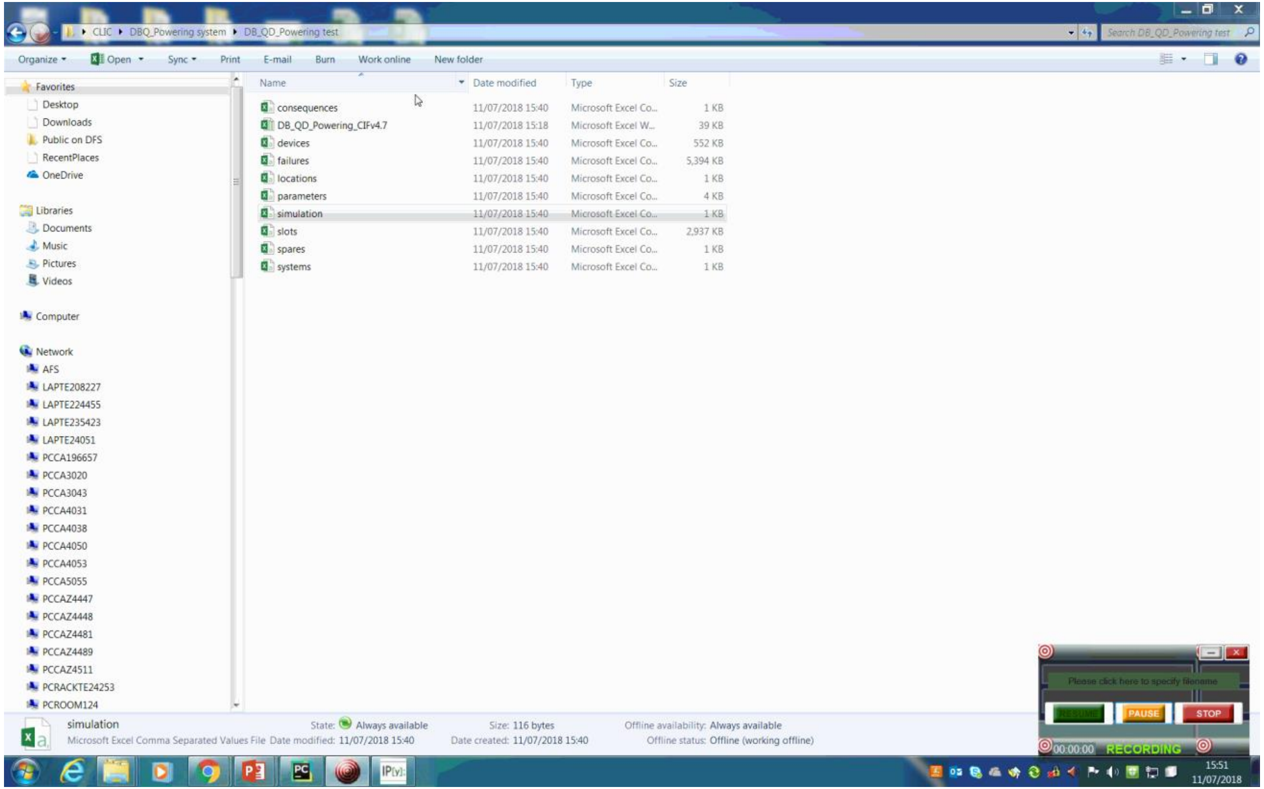
Phase definition

Facility	Phase	Duration	PhaseType	PhaseGroup	Parent OP Phase	Next Default phase	DownRepairPolicy
CLIC	OP	2400	Operation				1

Modelling the CLIC DB Quads powering in the Common Input Format

Translator and availability simulation

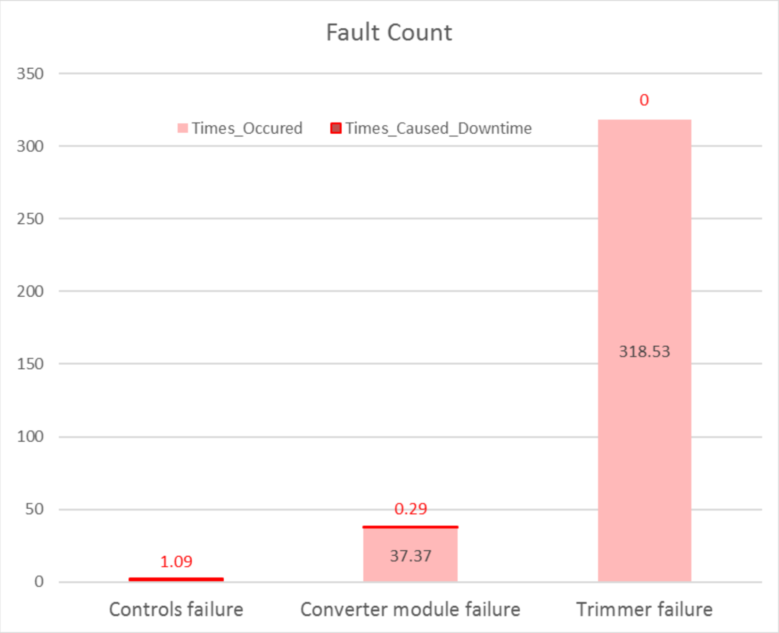




Modelling the CLIC DB Quads powering in the Common Input Format

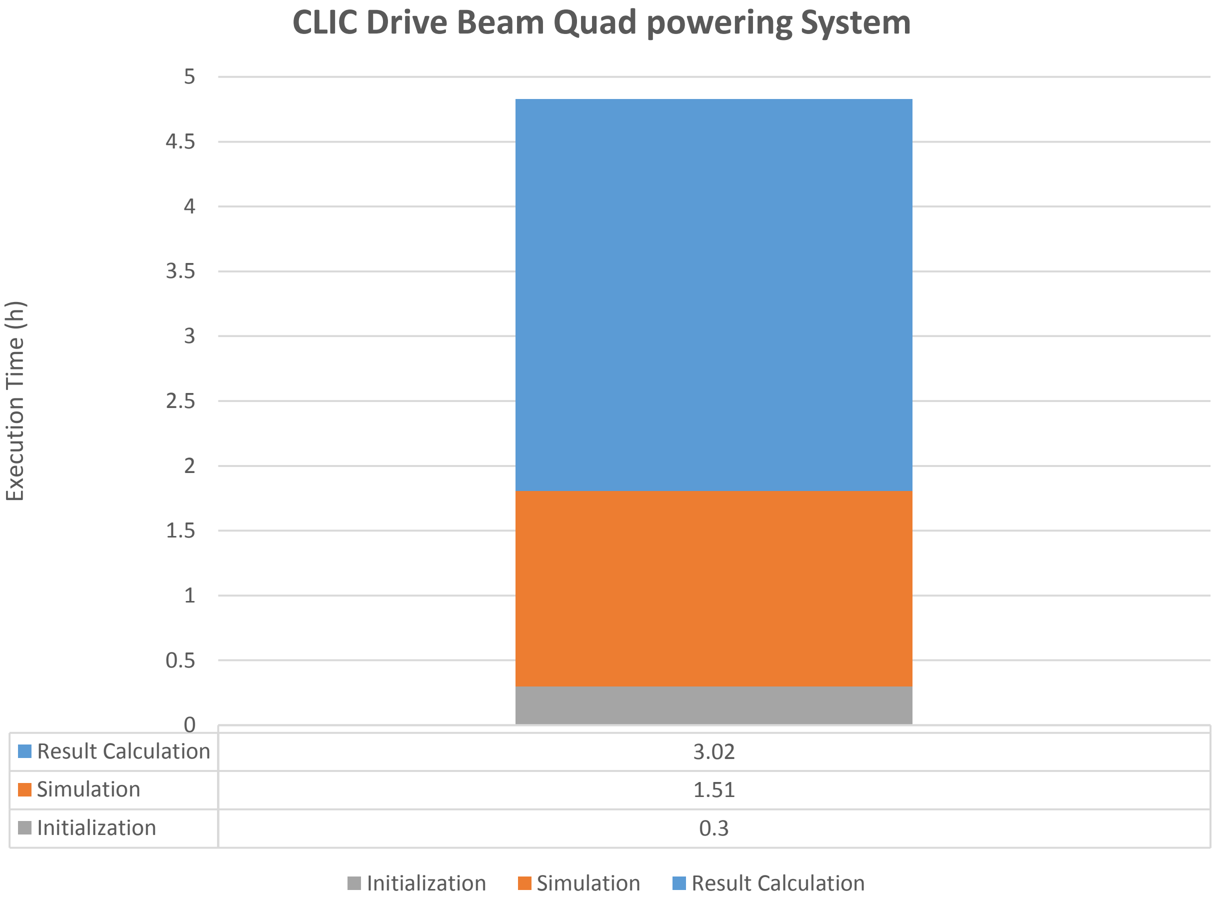
Results

Availability	Times Down	Uptime (days)	Downtime (h)	Standard deviation	MTTR (h)	MTBF (h)
99.7%	**1.38**	99.8	5.52	0.001	**4**	**1735**



Performance

[The CLIC Drive Beam Quad powering System]



Result Calculation	3.02
Simulation	1.51
Initialization	0.3

Analyzing the execution with cProfile:

1. 44689 components
2. 50 Simulations of 2400h
3. Results highly depends on model
4. ≈ 7600 LOC

Future improvement should focus on:

- Turn off log (implemented, measurements to be done)
- Multithread the simulation
- During result calculation, most time is spent calculating Availability graph (homemade solution) – solve by using matrixes in numba or numpy

Repository + Outlook

AvailSim repository & versioning policy

- CERN GitLab repository:
  - https://gitlab.cern.ch/mmotyka/AvailSim3 (private)
  - Source code + README for installation/software explanation
  - **Master branch** --> latest stable version
  - **Developer branch** --> regular minor changes
  - **Code standard** --> PEP8 (most popular, widely used)

- Versioning policy & changelog:
  - For every new functionality: v0.1.x+1 (latest is v0.1.8)
    - **[SoftwareVersion]:** dd-mm-yyyy
    - **Added:** New features
    - **Changed:** Changes in existing functionality
    - **Removed:** Deprecated features in the release
    - **Fixed:** Bug fixes

Brief Outlook

- Extension to cycling machines (ongoing)
- Continuous (rather than discrete) variation of accelerator parameters
- Performance improvements
- Collaboration with MPE-MS on maintainability/documentation

Thank you for your attention!