Modeling Software Aging and Rejuvenation Beyond the Enabling Restriction

Laura Carnevali[1] Marco Paolieri[2] Benedetta Picano[1] Riccardo Reali[1]
Leonardo Scommegna[1] Roberto Verdecchia[1] Enrico Vicario[1]

[1]Dept. of Information Engineering, University of Florence
[2]Dept. of Computer Science, University of Southern California

QualIta'23 - June 2023

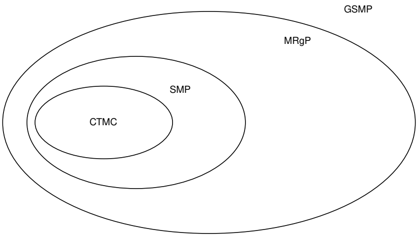
- this is about:
  - Stochastic models of software aging and rejuvenation
  - How class of the underlying stochastic process changes the expressivity
  - What are the consequences of expressivity change
  - Extension from MRgP under enabling restriction ...
  - ... by creating models in the class of MRgP under bounded regeneration

Sofware Aging & Rejuvenation Systems

- **Systems of interest** : long running hardware/software systems
- **Software Aging**: gradual shift from a robust state to degraded one
- **Rejuvenation**: restoration to the robust state through a **proactive maintenance**
    - **Advantage**: prevent failures
    - **Drawback**: implies a cost (downtime increment)
- **Rejuvenation Schedule**: plan when to perform the rejuvenation:
    - **Time-Triggered**: periodic rejuvenation
    - **Event-Based**: relying on diagnostic tests

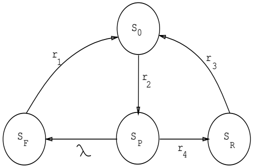
Software Aging & Rejuvenation Model

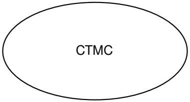
- Quantitative models analyse:
  - Effect of **aging** in the system
  - How **rejuvenation** impacts on the failure rate
  - Find the optimal rejuvenation **schedule** taking into account the **trade-off** between reliability and availability
- Differ in:
  - Available **model specification** approaches
  - Class of the underlying **stochastic process**



Continuous Time Markov Chain (CTMC)

- Expressivity:
  - **Markov condition**: always satisfied
  - **Sojourn Time**: only Exponential (memoryless) distributions
  - **Transition Selection**: time-independent random switch
  - Not good news for expressivity
- Specification:
  - Direct representation of the stochastic process
  - Higher level formalisms (SPN)

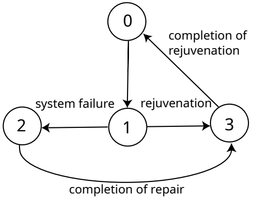


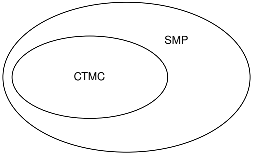


Model from: *Huang, Kintala, Kolettis, Fulton " Software rejuvenation: Analysis, module and applications", FTCS 1995*

Semi-Markov Processes (SMP)

- Expressivity:
  - **Markov Condition**: satisfied at the beginning of each sojourn time
  - **Sojourn Time**: GEN distributed durations
  - **Transition Selection**: based on a time-dependent random switch
  - Loses memory at each transition
- Specification:
  - Direct representation of the stochastic process

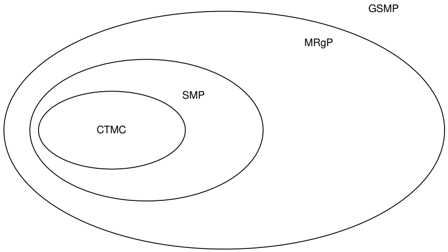




Model from: *Dohi, Goseva-Popstojanova, Trivedi "Statistical non-parametric algorithms to estimate the optimal software rejuvenation schedule", PRDC 2000*

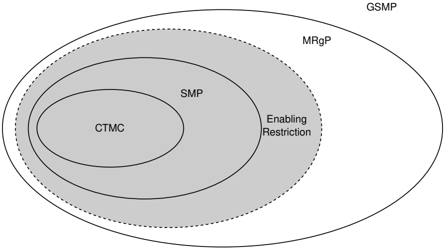
Markov Regenerative Processes (MRgP)

- Expressivity:
  - **Markov Condition**: eventually satisfied *w.p.1* at some point: the **regeneration**
  - **Sojourn Time**: GEN distributed durations
  - **Transition Selection**: based on the states and the time spent on them, since the last regeneration point
  - Represent activities overlapping their durations in epochs delimited by regenerations
- Specification:
  - Direct representation of the stochastic process
  - Usually expressed through higher level formalisms (SPN)



Modeling SAR Systems

- SAR systems literature deepens numerical approach with both **CTMC** and **SMP**
- Full-fledged MRgP models are **hard to analyse** numerically
- No general technique to built the process kernels
- **MRgP under enabling restriction**:
    - **Consolidated** numerical methods to analyse SAR
    - Basically, **at most one** GEN distribution fireable in any state

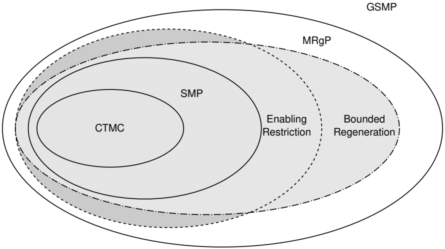


Consequences of the Enabling Restriction

- **Enabling Restriction:**
  - The single GEN distribution is usually (necessarily) spent for the **rejuvenation timer**
  - Distributions with finite coefficient of variation would result in **substantial approximation**
  - Remaining durations can be modeled **only as EXP**
- **Limits:**
  - **No concurrent GEN timers** can be represented
  - Durations can only fit the **first moment** from observational data

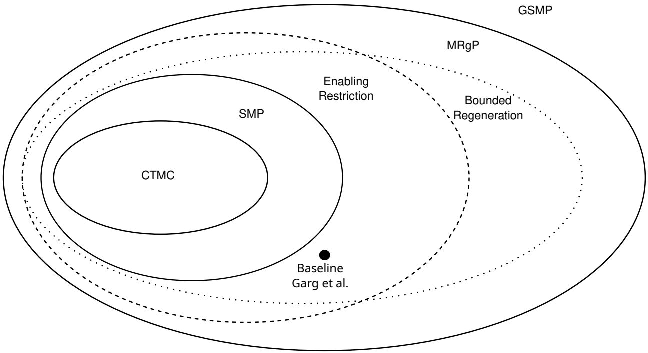
A Step Further, MRgPs Under Bounded Regeneration Restriction

- **Bounded Regeneration**:
  - Durations can be **DET** or **exponomial** distributions
  - **Bounded supports** distribution also allowed
  - **Constraint**: regeneration always reached within a **bounded** number of events
- **Stochastic State Classes**:
  - **Numerical solution** of model under bounded regeneration restriction
  - Implemented in the **Sirio Library** of the **Oris Tool**.[1]

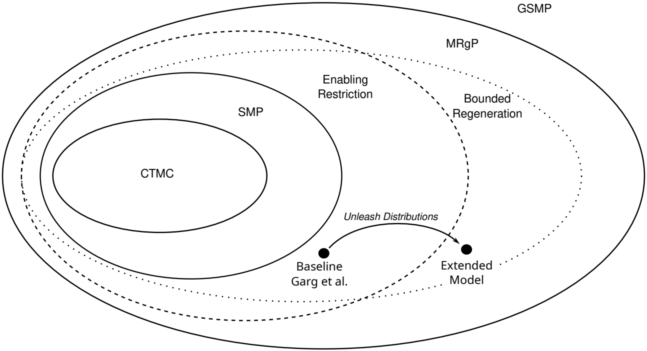


[1] *Paolieri, Biagi, Carnevali, Vicario, "The ORIS Tool: Quantitative Evaluation of Non-Markovian Systems" TSE 2021*

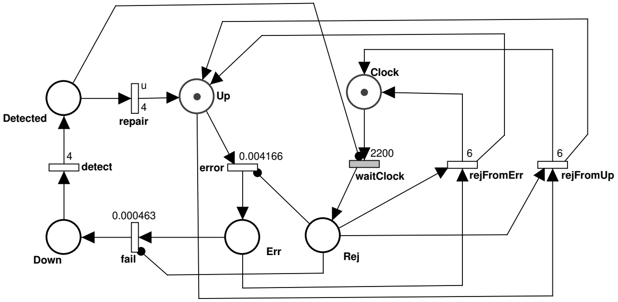
Starting Point



First of Two Steps Ahead, Unleashing Distributions



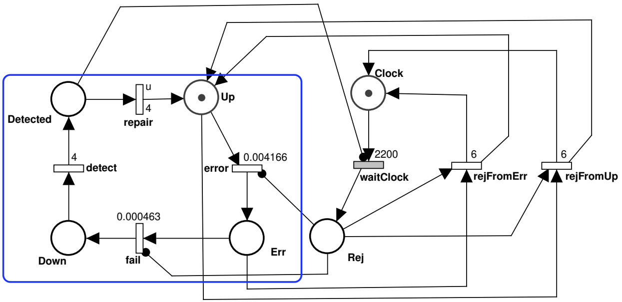
The SAR Model Baseline



- Based from the seminal model of *Garg et al.* [2]
- **Time-triggered** Rejuvenation with DET timer
- Tweaked to make it more complex
- All the duration except timer are EXP

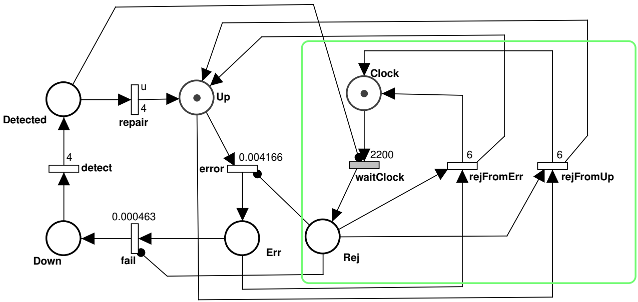
[2] *Garg, Puliafito, Telek, Trivedi "Analysis of software rejuvenation using Markov regenerative stochastic Petri net", ISSRE 1995*

The SAR Model Baseline - The System



- System starts in **robust state** ("`Up`") and eventually goes in **error** ("`Err`" place)
- From erroneous state system may **fail** ("`Down`" place)
- Failure detection time is **not negligible** ("`Down`" and "`Detected`" places)

The SAR Model Baseline - The Rejuvenation Mechanism

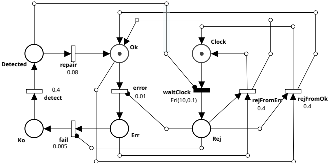


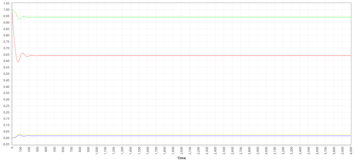
- Rejuvenation mechanism **runs concurrently** with the actual system
- When triggered (“`Rej`” place), it **restores** the system in the `Up` place
- Possible in both “`Up`” and “`Err`” places

Effects of the Model Expressiveness on the Analysis Results

WaitClock
Erl(10,0.1)

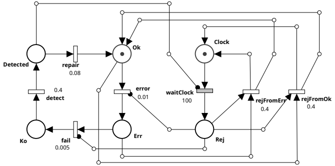
CTMC
(Erl as sequence of EXP)

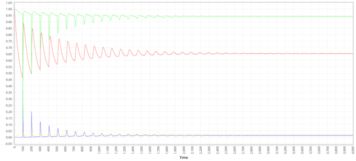




WaitClock
DET 100

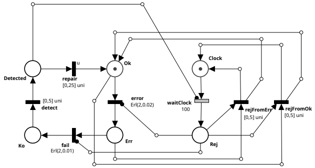
MRgP
Enabling
Restriction

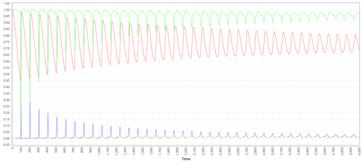




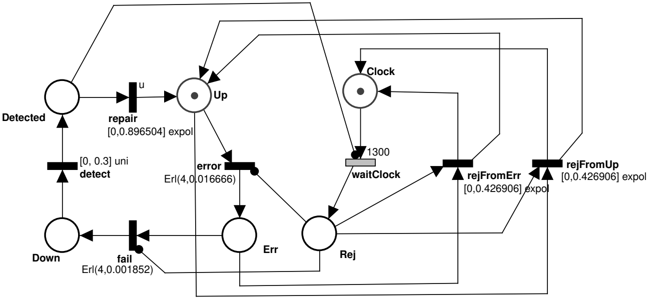
WaitClock
DET 100

MRgP
Bounded
Regeneration





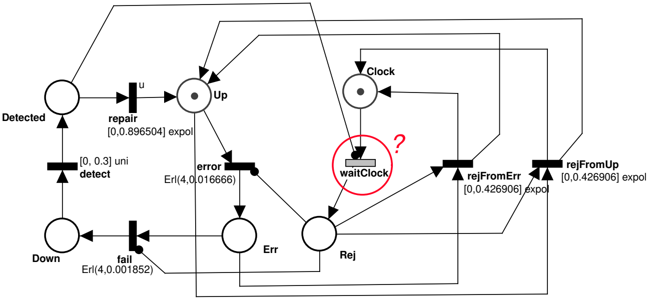
Extension Beyond Enabling Restriction



- The **same** structure of concurrence
- **GEN distributed** Durations
- Stochastic parameters maintain the **same expected values** of the baseline

Model from: *Carnevali, Paolieri, Reali, Scommegna, Vicario, "A Markov Regenerative Model of Software Rejuvenation Beyond the Enabling Restriction", WoSAR 2022*

The Optimal Rejuvenation Period



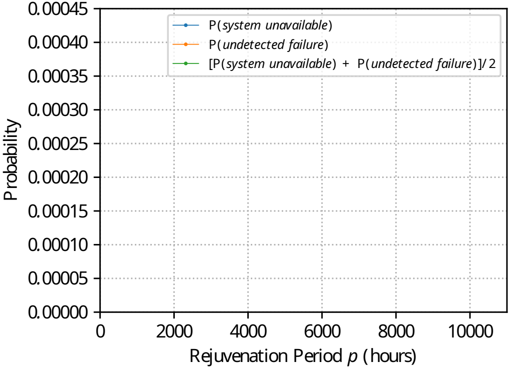
- **Aim**: Finding the best rejuvenation period
- A clock time aware of the reliability-availability **trade-off**
- The optimal value of the `Wait clock` transition

Finding the Optimal Rejuvenation Period

- **Steady-state** metrics are the core of the software aging analysis
- **Unavailability** $\bar{a}(p)$: quantifies the occurrence of states in which the system can not provide the service
- **Undetected failures** $\bar{r}(p)$: detect a reliability quality of the system
- **Optimal rejuvenation period** p^***:** selected minimizing:

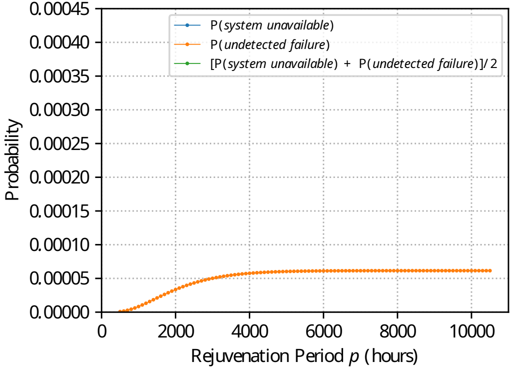
$$p^* = \arg\min_p \left[\frac{\bar{a}(p) + \bar{r}(p)}{2} \right]$$

Optimal Rejuvenation with Bounded Regeneration



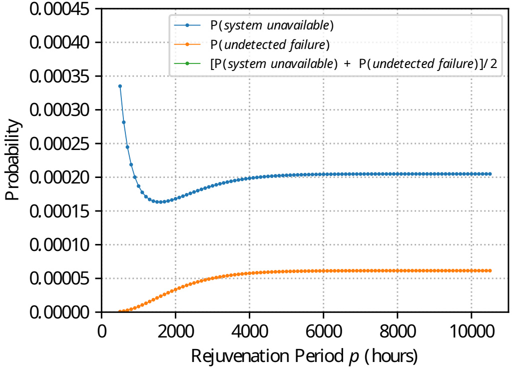
- $\bar{r}(p)$: increase with the rejuvenation period p
- $\bar{a}(p)$: initial decrease with subsequent increase
- p^*: 1300 hours with $\left[\frac{\bar{a}(p)+\bar{r}(p)}{2}\right] = 0.000091$

Optimal Rejuvenation with Bounded Regeneration



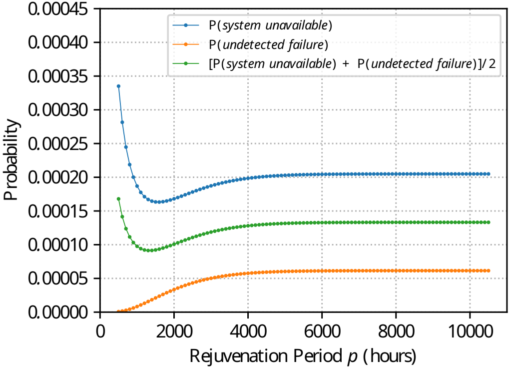
- $\bar{r}(p)$: increase with the rejuvenation period p
- $\bar{a}(p)$: initial decrease with subsequent increase
- p^*: 1300 hours with $\left[\frac{\bar{a}(p)+\bar{r}(p)}{2}\right] = 0.000091$

Optimal Rejuvenation with Bounded Regeneration



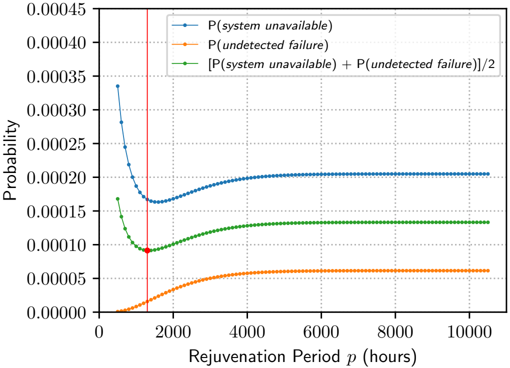
- $\bar{r}(p)$: increase with the rejuvenation period p
- $\bar{a}(p)$: initial decrease with subsequent increase
- p^*: 1300 hours with $\left[\frac{\bar{a}(p)+\bar{r}(p)}{2}\right] = 0.000091$

Optimal Rejuvenation with Bounded Regeneration



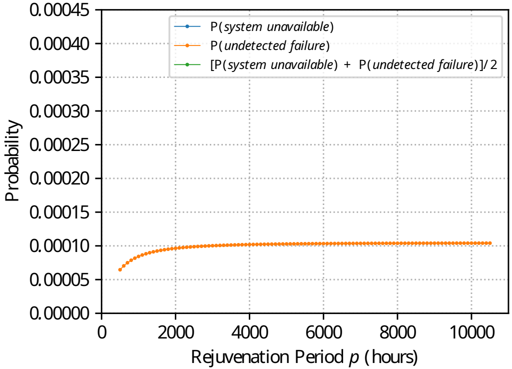
- $\bar{r}(p)$: increase with the rejuvenation period p
- $\bar{a}(p)$: initial decrease with subsequent increase
- p^*: 1300 hours with $\left[\frac{\bar{a}(p)+\bar{r}(p)}{2}\right] = 0.000091$

Optimal Rejuvenation with Bounded Regeneration



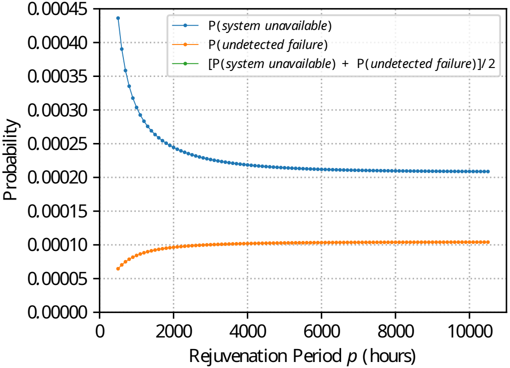
- $\bar{r}(p)$: increase with the rejuvenation period p
- $\bar{a}(p)$: initial decrease with subsequent increase
- p^*: 1300 hours with $\left[\frac{\bar{a}(p)+\bar{r}(p)}{2}\right] = 0.000091$

Optimal Rejuvenation with Enabling Restriction



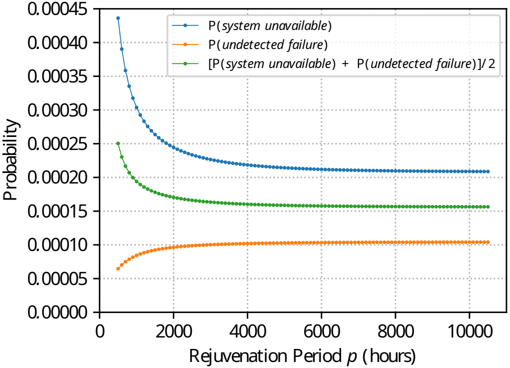
- $\bar{r}(p)$: monotonic trend increasing
- $\bar{a}(p)$: monotonic trend decreasing
- p^*: cannot be assigned through the equation
- Selected p optimal as 2200 to anticipate the mean time of error detection (2300 hours)

Optimal Rejuvenation with Enabling Restriction



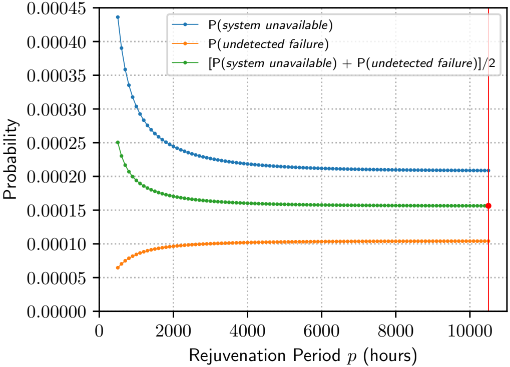
- $\bar{r}(p)$: monotonic trend increasing
- $\bar{a}(p)$: monotonic trend decreasing
- p^*: cannot be assigned through the equation
- Selected p optimal as 2200 to anticipate the mean time of error detection (2300 hours)

Optimal Rejuvenation with Enabling Restriction



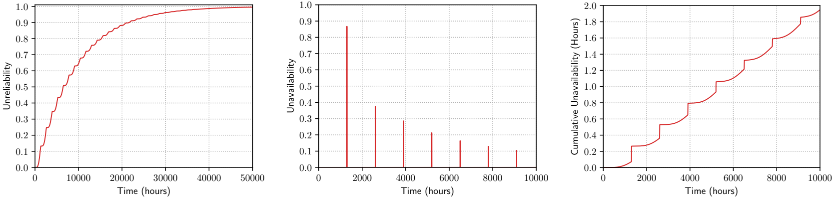
- $\bar{r}(p)$: monotonic trend increasing
- $\bar{a}(p)$: monotonic trend decreasing
- p^*: cannot be assigned through the equation
- Selected p optimal as 2200 to anticipate the mean time of error detection (2300 hours)

Optimal Rejuvenation with Enabling Restriction



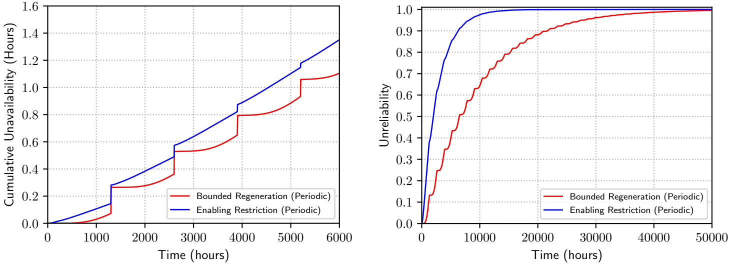
- $\bar{r}(p)$: monotonic trend increasing
- $\bar{a}(p)$: monotonic trend decreasing
- p^*: cannot be assigned through the equation
- Selected p optimal as 2200 to anticipate the mean time of error detection (2300 hours)

Transient Evaluation of the Bounded Regeneration Model



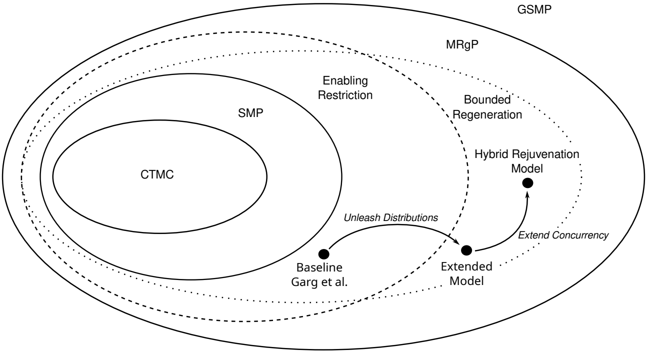
- **Unreliability** (left): converges to 1 after 50000 hours
- **Unavailability** (center): peaks around the multiples of p^* (1300 hours)
- **Cumulative unavailability** (right): sharp increase around the multiples of p^* (smoothed as time advances)

Transient Metrics Comparison



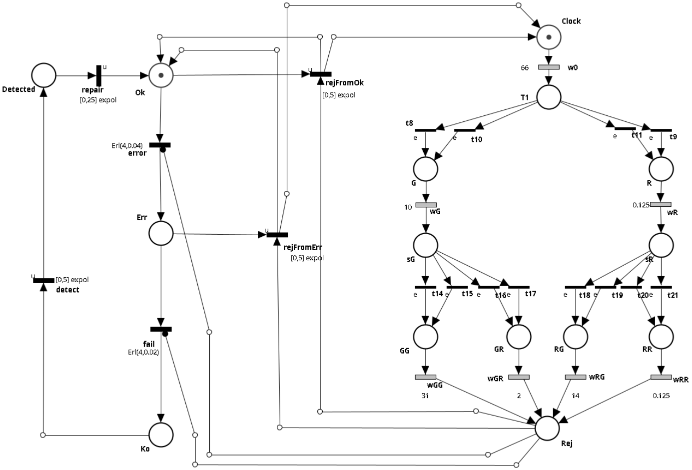
- **Unavailability**: bounded regeneration shows better performance
- **Unreliability**: enabling restriction make the model converge faster
- Type of distributions affect significantly the transient behavior

Bounded Regeneration: Beyond Parameters Fitting



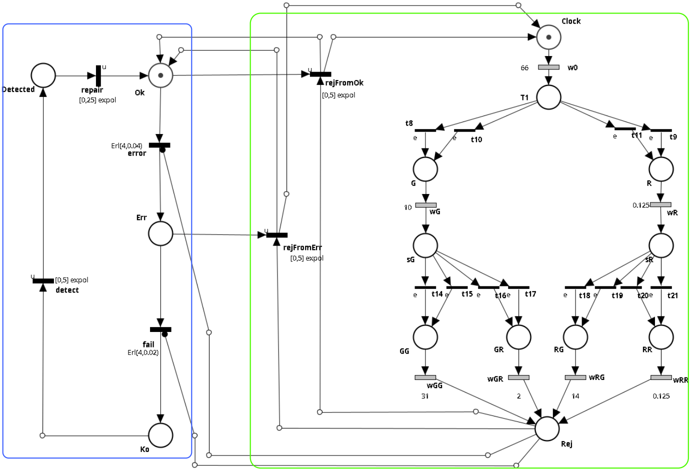
- Using expressiveness gain **not only** to fit more precisely data
- Structure of concurrency modification to represent **hybrid rejuvenation policies**

Combining Event-Based and Time-Triggered Rejuvenation



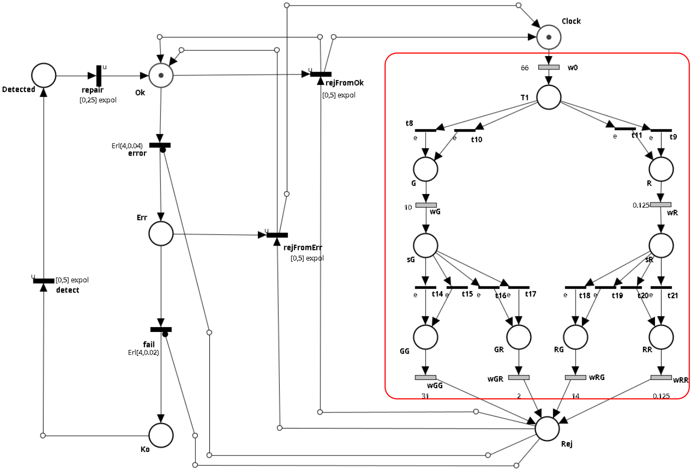
- Structure of concurrency extended to include diagnostic process
- Same system behavior and rejuvenation effects
- Diagnostic tests executed during rejuvenation period

Combining Event-Based and Time-Triggered Rejuvenation



- Structure of concurrency extended to include diagnostic process
- Same system behavior and rejuvenation effects
- Diagnostic tests executed during rejuvenation period

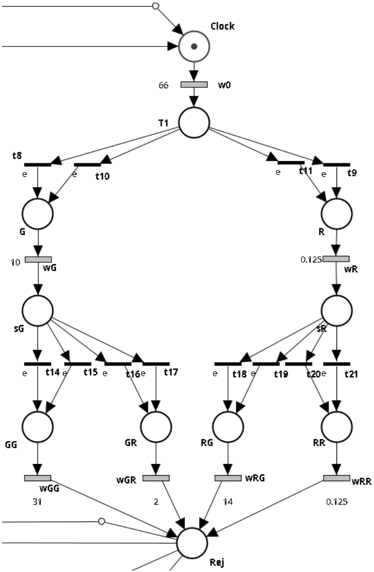
Combining Event-Based and Time-Triggered Rejuvenation



- Structure of concurrency extended to include diagnostic process
- Same system behavior and rejuvenation effects
- Diagnostic tests affect rejuvenation times

Patameters of the Rejuvenating System

- **Sensitivity and Specificity**: modeled through `t` transitions
- **Defer Times**: modeled through `w` transitions (DET distributions)
- Customizable **number of tests** (2 in the example)



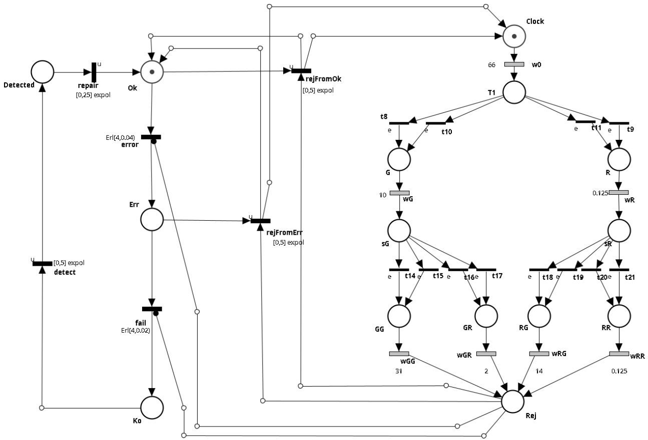
Parametrization of Times of Diagnostic Tests

- **Aim**: Identifying optimal defer times maintaining a uniform failure density over time:

$$\frac{P(KOwithinT|Observations)}{T} < \epsilon$$

- Deferring rejuvenation of T units of time implies a failure probability of $T * \epsilon$
- The defer time is chosen to maintain a uniform first-failure rate

Parametrization Algorithm



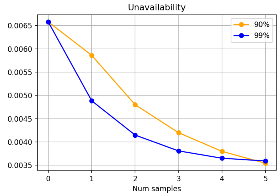
- Within a Single Rejuvenation Epoch
- **Incremental procedure** to calculate the defer time T_{wK_n}
- K_n is a sequence of n words in $\{G, R\}$
- T_{wK_n} is calculated on top of $T_{wK_{n-1}}$

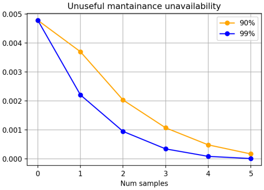
Parametrization with Sensitivity and Specificity of 0.9 and $\epsilon = 10^{-5}$

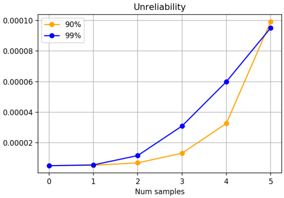
- Previous defer times are the **same**
- Test result is **more relevant** than the elapsed time
- A negative test result leads to **immediate retesting**
- A positive test, even after multiple negative tests, can result in a **high deferment**

Test Sequence	1 Test	2 Test	3 Test
*W*0	66.0	66.0	66.0
WG	12.0	12.0	12.0
WR	0.125	0.125	0.125
WGG		29.0	29.0
WGR		1.0	1.0
WRG		15.0	15.0
WRR		0.125	0.125
WGGG			24.0
WGGR			0.125
WGRG			45
WGRR			0.125
WRGG			57
WRGR			0.125
WRRG			0.17
WRRR			0.125

Quantities of Interest







- **Unavailability** (left): decreases as the number of tests increases
- **Unuseful unavailability** (center): same behavior of unavailability
- **Unreliability** (right): increases with the number of tests

Discussion and Future Directions

- **Aim of the Work:**
  - Demonstrating that processes in the class of MRgP under bounded regeneration restriction provide an expressivity gain
  - Modeling a Hybrid Rejuvenation Strategy
- **Ongoing Direction:**
  - Optimal Parametrization of defer times