The Timing Diagram Editing and Verification Method

BINP, Russia

G. A. Fatkin , A. I. Senchenko

Budker Institute of Nuclear Physics, Novosibirsk; Novosibirsk State University, Novosibirsk

E-mail: G.A.Fatkin@inp.nsk.su

Introduction

Preparation and verification of the timing diagrams for modern complex facilities with diversified timing systems is a difficult task. A mathematical method for convenient editing and verification of the timing diagrams is presented. This method is based on systems of linear equations and linear inequalities. Every timing diagram has three interconnected representations: a textual equation representation, a matrix representation and a graph (tree) representation. A prototype of the software using this method was conceived in Python. This prototype allows conversion of the timing data between all three representations and its visualization.

System of Linear Equations

Example of a time diagram for traffic light

$$\begin{cases} t_{green} = 0 \\ t_{yellow} = t_{green} + 5 \\ t_{red} = t_{yellow} + 5 \end{cases}$$

$$\downarrow$$

$$\begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 5 \\ 5 \end{pmatrix} \qquad v = \begin{pmatrix} "Green" \\ "Yellow" \\ "Red" \end{pmatrix}$$

In general case, the time diagram of an installation could be described by a system of linear equations which could be written in the following form:

$$At = b,$$

where t - the vector of the event times, A - the system matrix, and b - the delay vector. We also need a vector v comprised of the event names.

For the time diagram to be correct, the matrix A has to be invertible, in this case the system is solvable. If matrix A is not invertible, there is an error in time diagram specification.

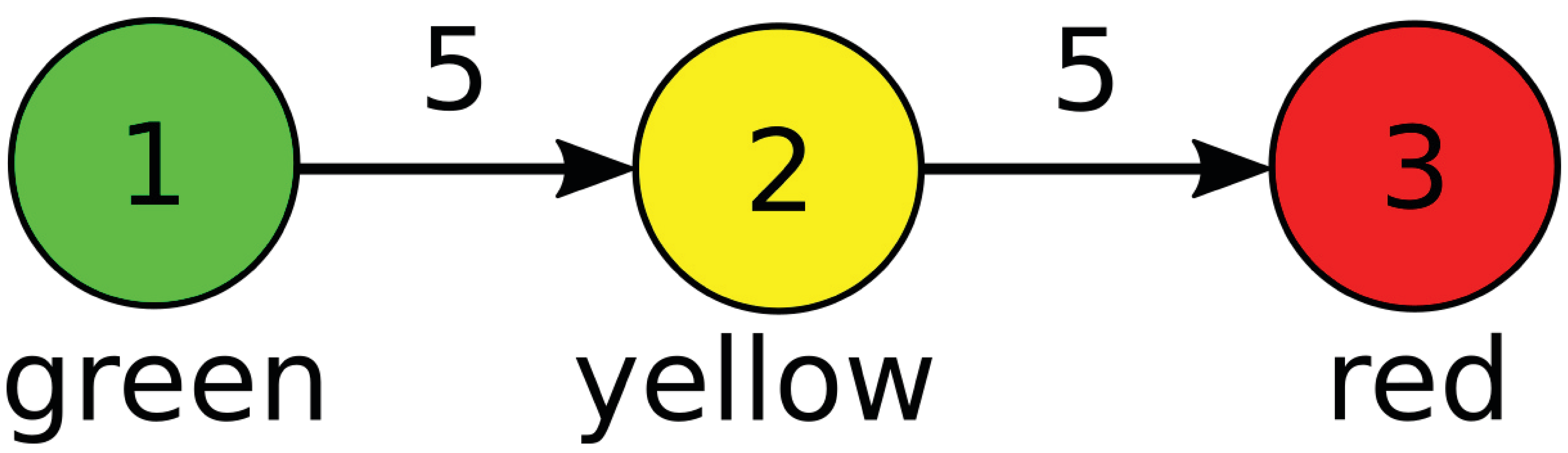
Events Graph

The matrix representation of time diagram is quite convenient for storage and computation of the event times, but it does not allow to visually explore the relationship of the events. Let us deduce the following matrix:

$$G = E - A,$$

where E - is an eye matrix of the same dimension as A.

We can note that the matrix G is an adjacency matrix of a directed graph. The vertex of a graph represents an event and an edge represents a relationship. It is logical to place the name of the event in the vertex, and to make the weight of an edge equal to the appropriate element of the vector b. Thus we get a representation of time diagram as a graph.



Time Constraints

It is usual to impose some constraints on the time diagram. Such constraints could be on the relative position of the events, total diagram duration, positioning event in some range, etc. All theese constraints could be represented as one or several linear inequalities.

By multiplying left and right parts by -1, one could easily convert the system to the following form:

$$Ct \geq d,$$

where C - matrix of inequalities, t - event times vector and d - constraints vector.

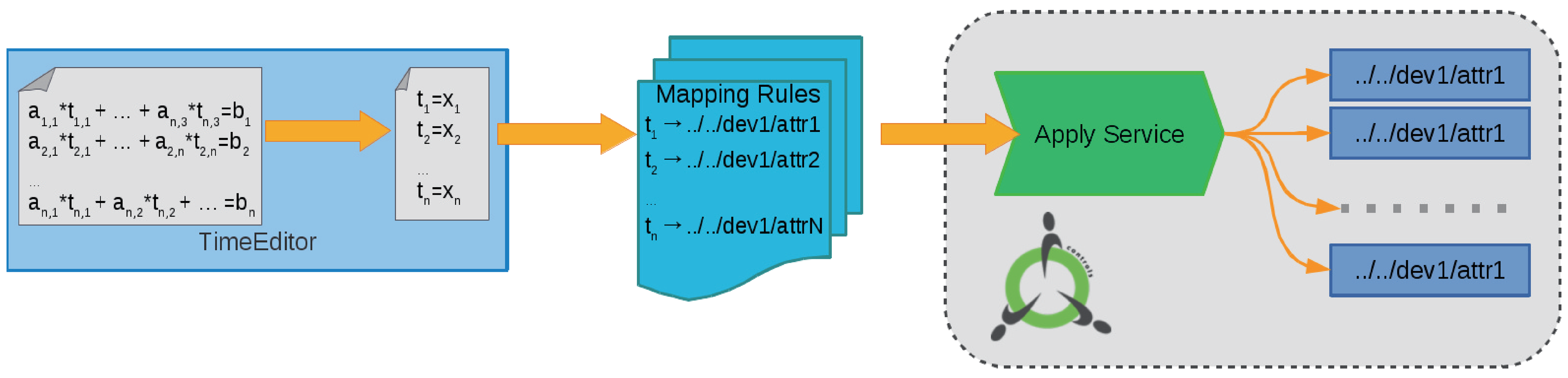
$$\begin{cases} 3 \leq t_1 \leq 6 \\ 2 \leq t_2 - t_1 \leq 5 \end{cases} \qquad \begin{pmatrix} 1 & 0 \\ -1 & 0 \\ -1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} \geq \begin{pmatrix} 3 \\ -6 \\ 2 \\ -5 \end{pmatrix}$$

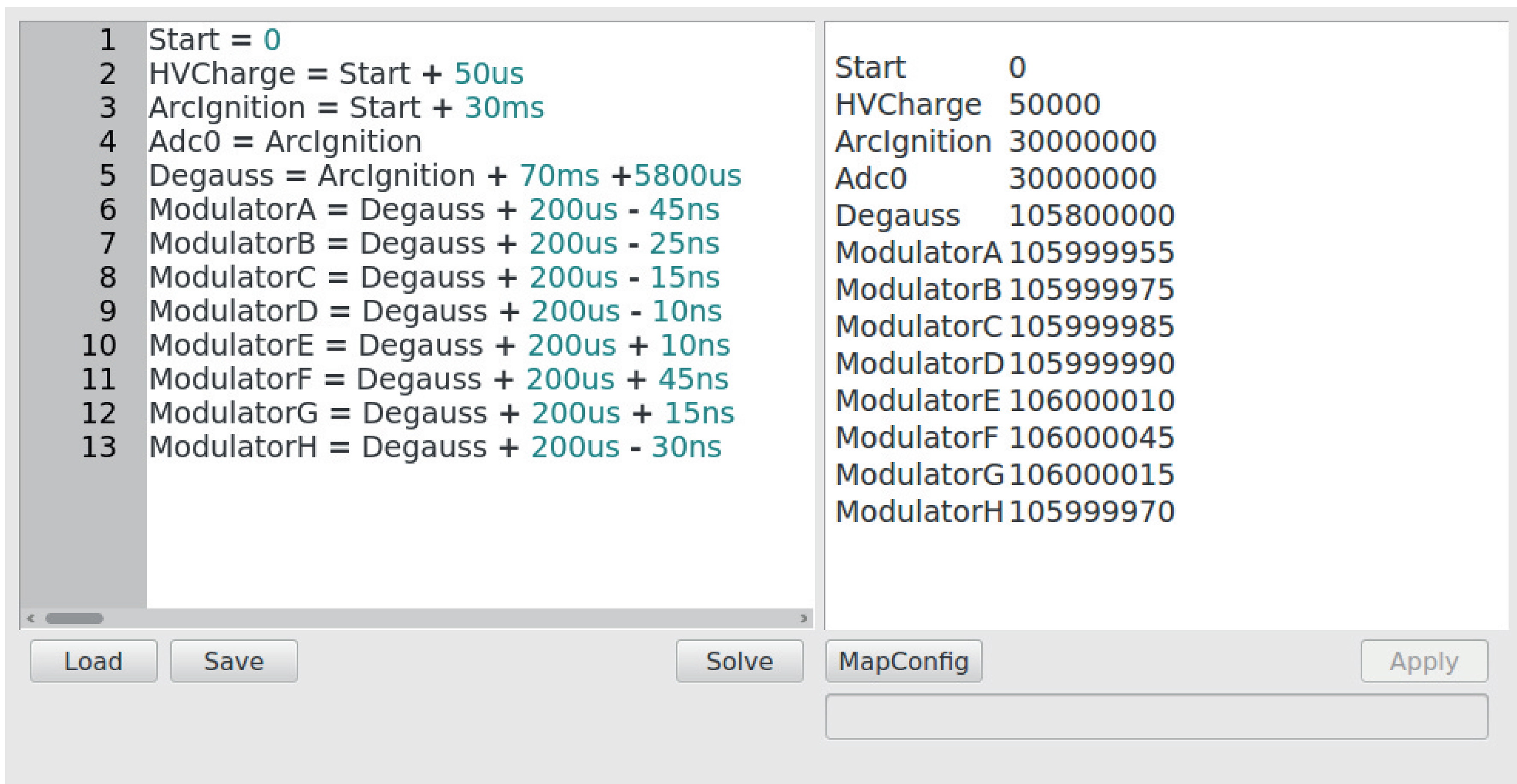
This problem is reduced to the classical problem of a linear programming being the problem of minimization of linear functional with some boundary conditions:

$$\min\{(c,y),\ Dy = e,\ y \geq 0\},$$

where we have to assume the objective function c = 0. To solve this problem we can use an algorythm using the simplex-method. The algorythm will either provide us the first vertex of the convex polyhedron, which is the edge of the solutions of the system of linear inequalities, or it will determine that the system is unsolvable.

Prototype Implementation





Future developments

We plan to create an export module from the solver for EPICS and CX-Server control systems and use it to create time diagrams on functioning BINP accelerators. We also plan to realize the check the diagram consistency using a system of linear inequalities. The ideal evolution of the program is the creation of an interactive instrument that allows: editing time diagrams in any of three forms: text representation, matrices or graph and plotting the events line of the time diagram.

ICALEPCS 2017 - TUPHA087