

Linguaggi dataflow

LabVIEW, NodeRed, Simulink



Andrea Sterbini – sterbini@di.uniroma1.it

Data-flow: interconnected functional units

Functional units connected by data-links

- links represent data exchanges (i.e. variables)
- they could be typed
- many data could be aggregated in a single BUS
- each unit can have a default GUI for testing

Inherently parallel

- linked units must run sequentially
- NON-linked units run in parallel (if all data is available)

You can build and reuse functional units

- you choose the granularity of your blocks
- you can build a block containing a sub-net

LabVIEW

Created by National Instruments to interact with digital data-acquisition and control systems

Modelled over the circuit design and testing metaphor

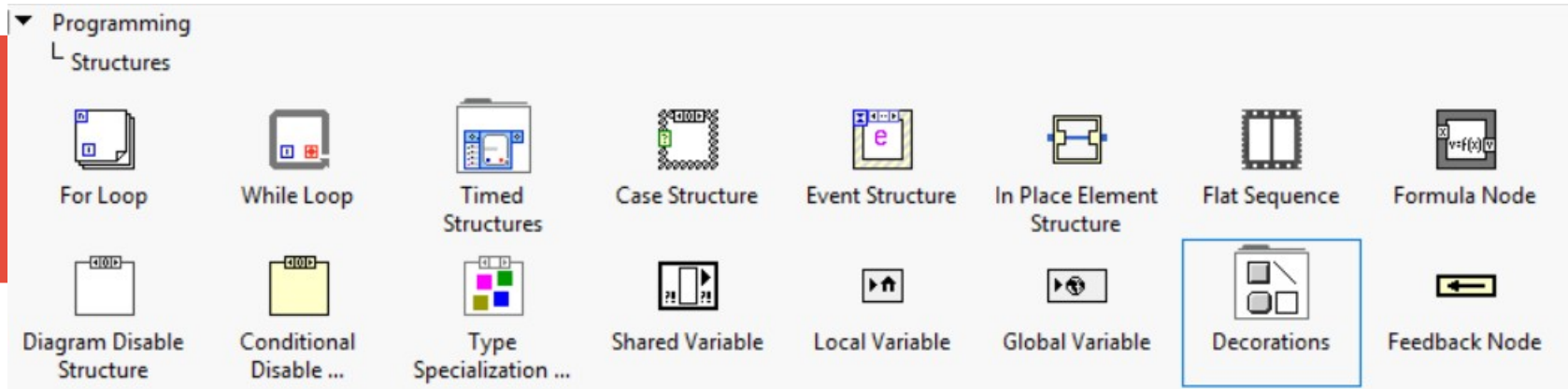
Each functional unit in the graphic language runs as soon all its input data are available

Multiple cores and threads are used to schedule the parallel execution of multiple active units

The programs are compiled into an intermediate “G” language
(but can also be compiled to native code)

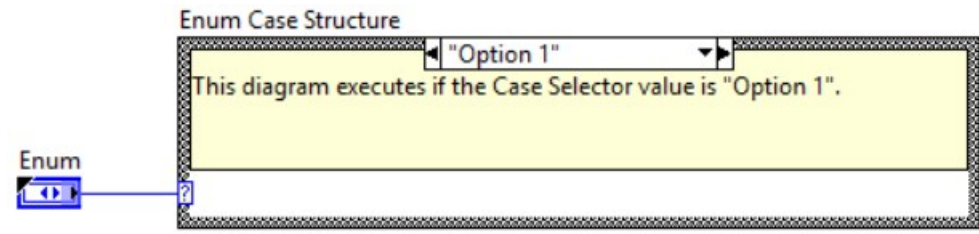
You normally (need to) add explanation boxes to document your ideas

LabVIEW



Loops, cases, if/then/else are represented as boxes

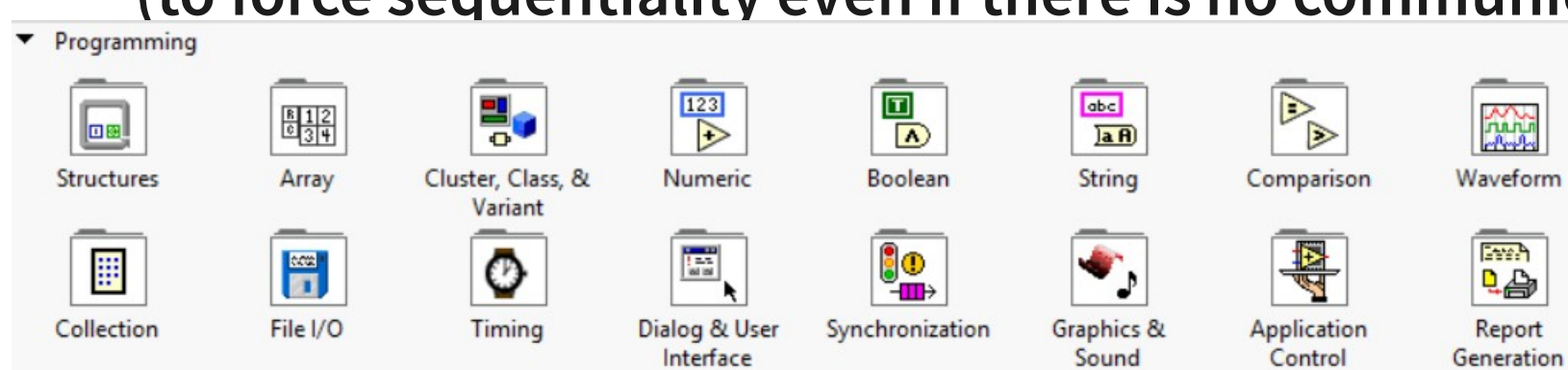
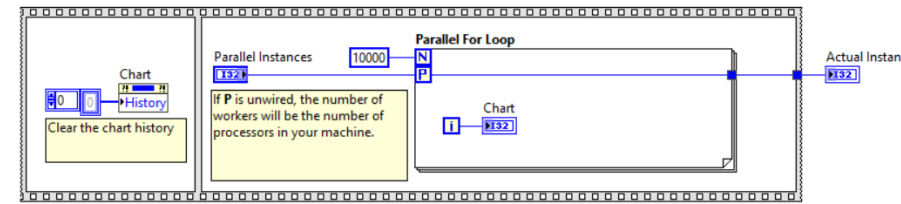
- a conditional input on the border control the loop/if/case
- the inner content is the conditioned block
- each case is a “page” of the box



Sequential parts are represented:

- as interconnected functional units
- or as boxes sharing one side

(to force sequentiality even if there is no communication)



Data-flow

LabView programming style

Data-flow

Visual construction of the data-flow diagram

Visual test of the diagram (all blocks have their GUI)

Inherently parallel (you just forget about sequentiality constraints)

Object-Oriented (classes)

Interaction with other systems:

- Function blocks for data math manipulation
- Code blocks for special algorithms
- Many libraries for Statistics, Signal analysis/manipulation, Math
- Many external libraries

Demo

DEMO

Data-flow

Node-RED (<http://nodered.org>)

Flow-based visual programming tool created initially by IBM for IoT

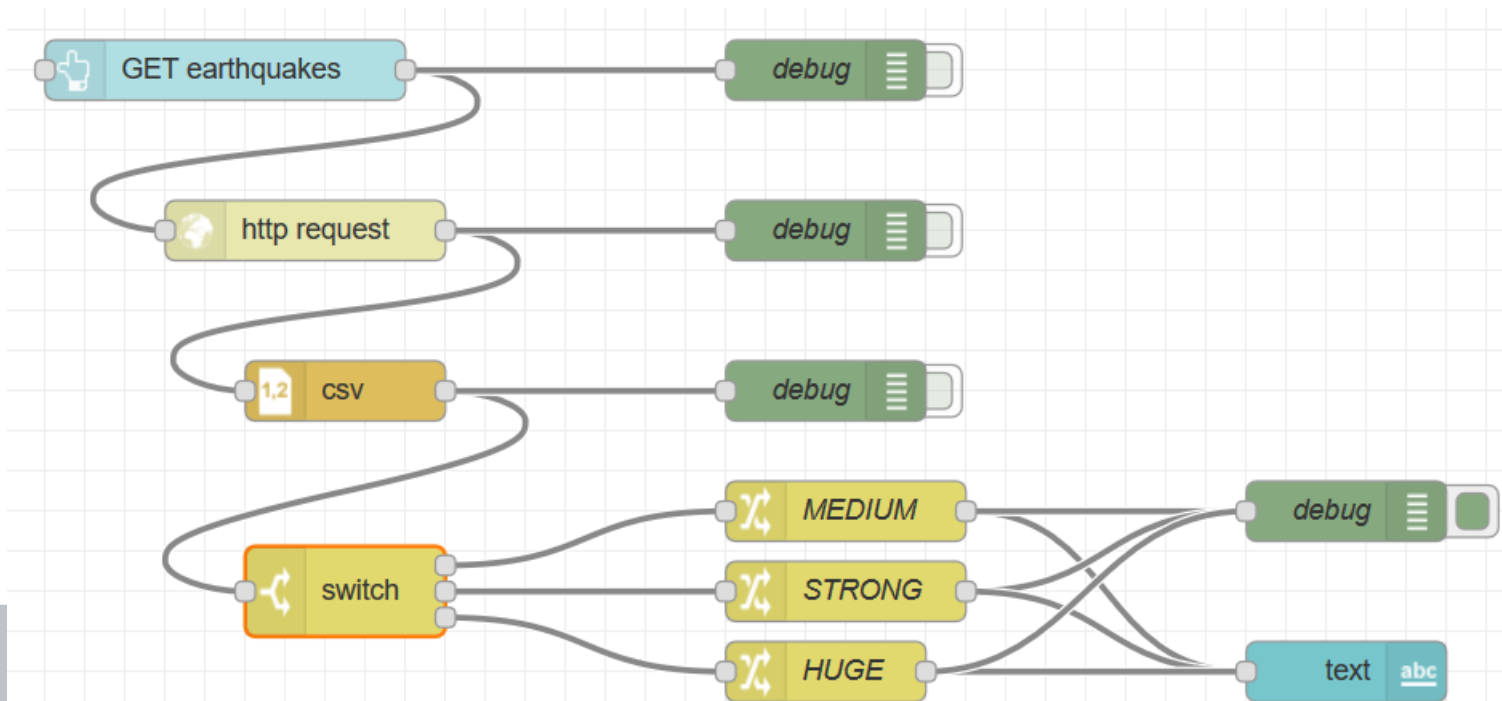
JavaScript functional units interconnected (installed with NPM)

Wires communicate JSON messages aggregating all data

- functional units get/add data to the messages
- messages can be split/joined into message sequences

Inherently
parallel

A node could
contain full
programs



Demo

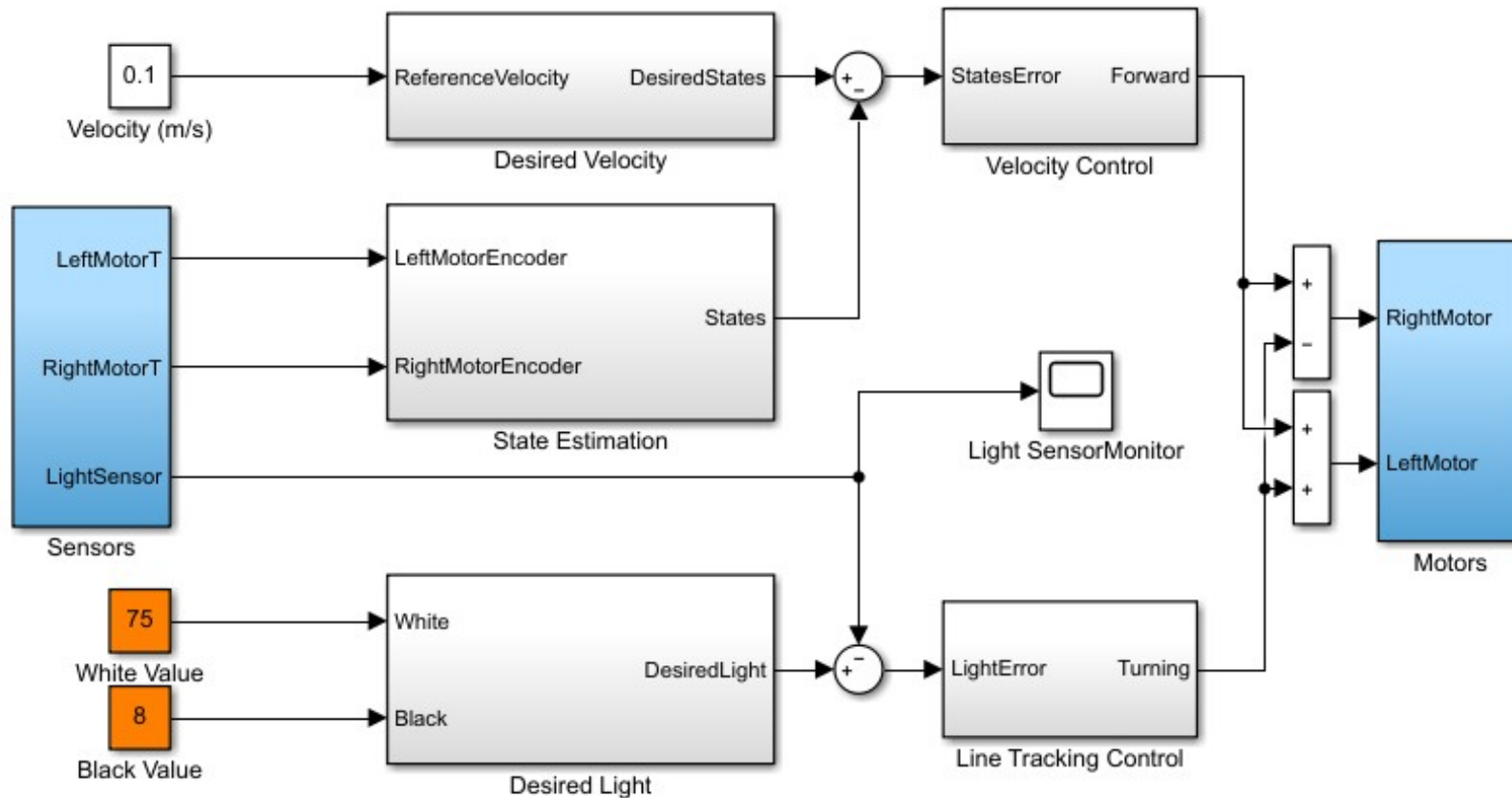
DEMO

Data-flow

Data-flow programming inside MatLab, very circuit-based

Line Tracking

Copyright 2014-2015 The MathWorks, Inc.



Demo

DEMO

Data-flow