Visual Programming of Content Processing Grid

Paolo Nesi

DISIT-DSI, Distributed Systems and Internet Technology Lab

Department of Systems and Informatics, University of Florence

nesi@dsi.unifi.it, paolo.nesi@unifi.it

http://www.dsi.unifi.it/~nesi, ***http://www.disit.dsi.unifi.it***

DMS 2009, San Francisco, USA

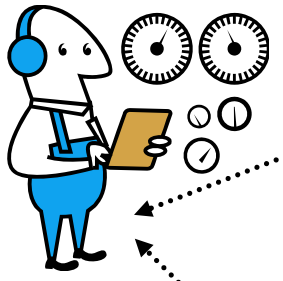
Overview

- Rationales
- Application domain
- General requirements for grid media processing
- AXCP overview
- Analysis of the GRID rules
- The visual language and model
- Applications
- Conclusions

Rationales

- **Content processing grid/parallel infrastructures are becoming every day more needed and diffuse,**
  - cloud computing.
- **Several requirements are stressed for those grids:**
  - Program Flexibility
  - Reconfigurability
  - Integration with workflow management systems
  - Visual programming
  - Large/huge number of functionalities
  - Integration of many functionalities from data processing to taking decision engine, semantic processing, etc.
  - Distributed debugging, Etc.
- **to solve a part of those problems a**
  - visual programming tool for media on grid processing has been defined and validated
  - Starting from the AXMEDIS grid and model, AXCP grid.

AXCP for empowering Media Portals



AXMEDIS Rule Editor

Few Ready to use rules

User and Device profile

Add info

Activate Rule

Content Databases

Scheduler

AXCP GRID

Personalized Content on Demand

Users

Distributor front end Server

Content: Search, Selection, Acquisition, Production, Adaptation, Transcoding, Formatting, Packaging, Protection, Publication and Licensing on Demand

AXMEDIS DRM

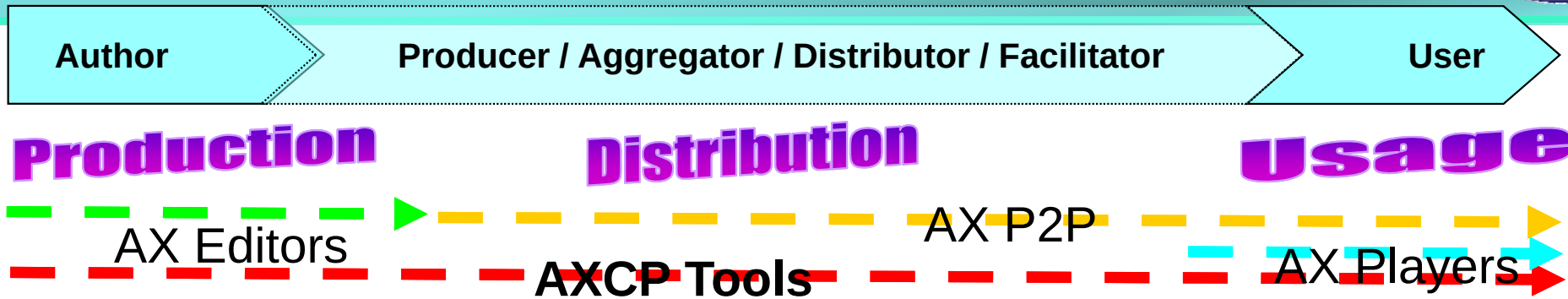
Other DRM

YouTube, ..

P2P, ...

APPLICATIONS: balancing, UGC, repurposing, publication, recommendations, integration with SN, redistribution, crawling

Enabling massive content production and processing

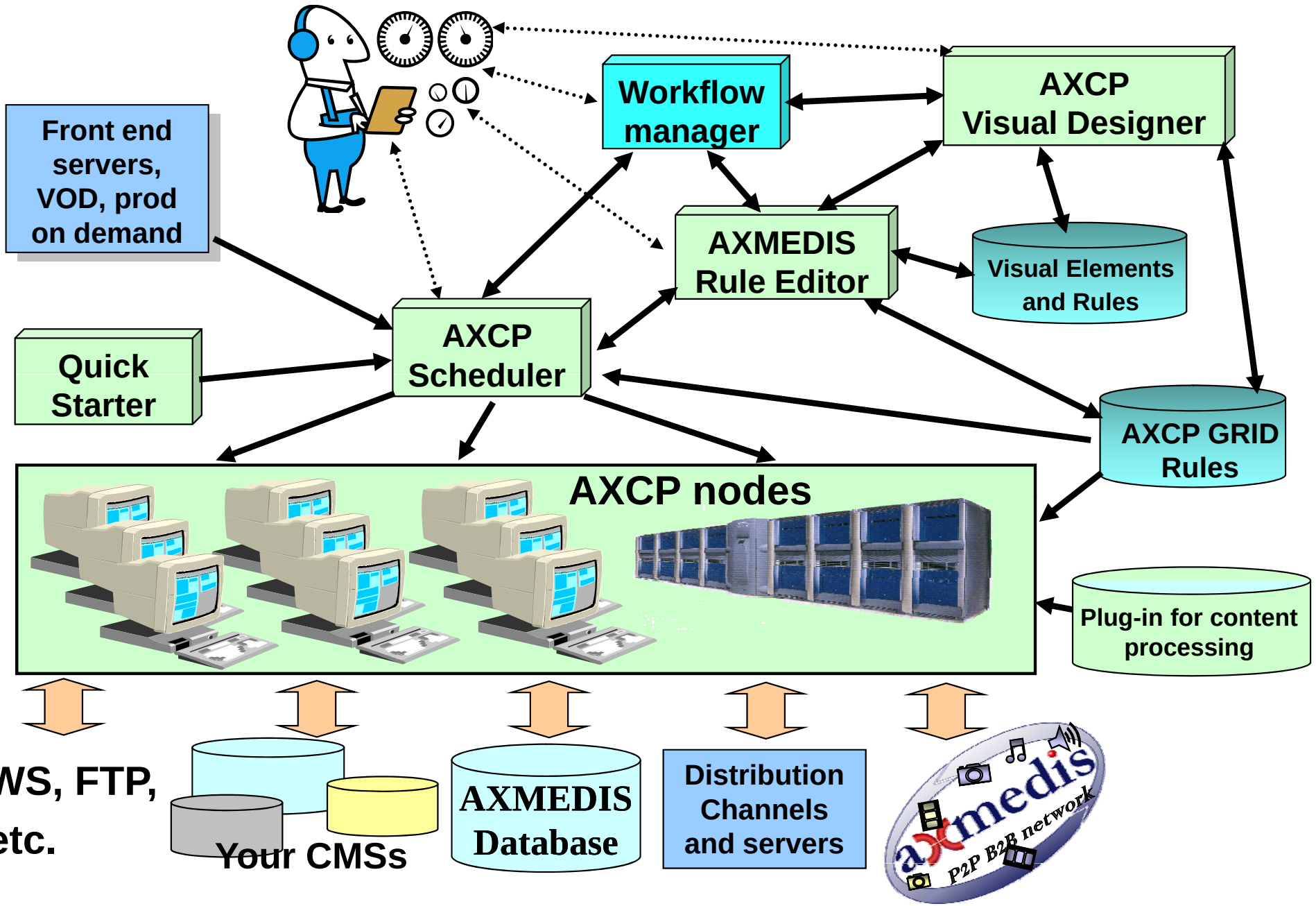


- **AXMEDIS GRID language and tools for**
  - Automated Content Ingestion and Gathering
  - Automated Content Query and Retrieval
  - Automated Content Load and Storage: databases and files
  - Automated Content and Metadata Processing, enrichment
  - Automated Content and Data processing
  - Automated Content Filtering and Repurposing
  - Automated Content Composition, Formatting
  - Automated Content Protection and Licensing
  - Automated Content Publication/Download on/from any channel
  - Automated Content Distribution via Multichannel
  - Automated Profiles management and processing
  - Automated Production of Content on Demand
  - **Automated semantic processing, recc, taking decisions, etc..**
  - .....

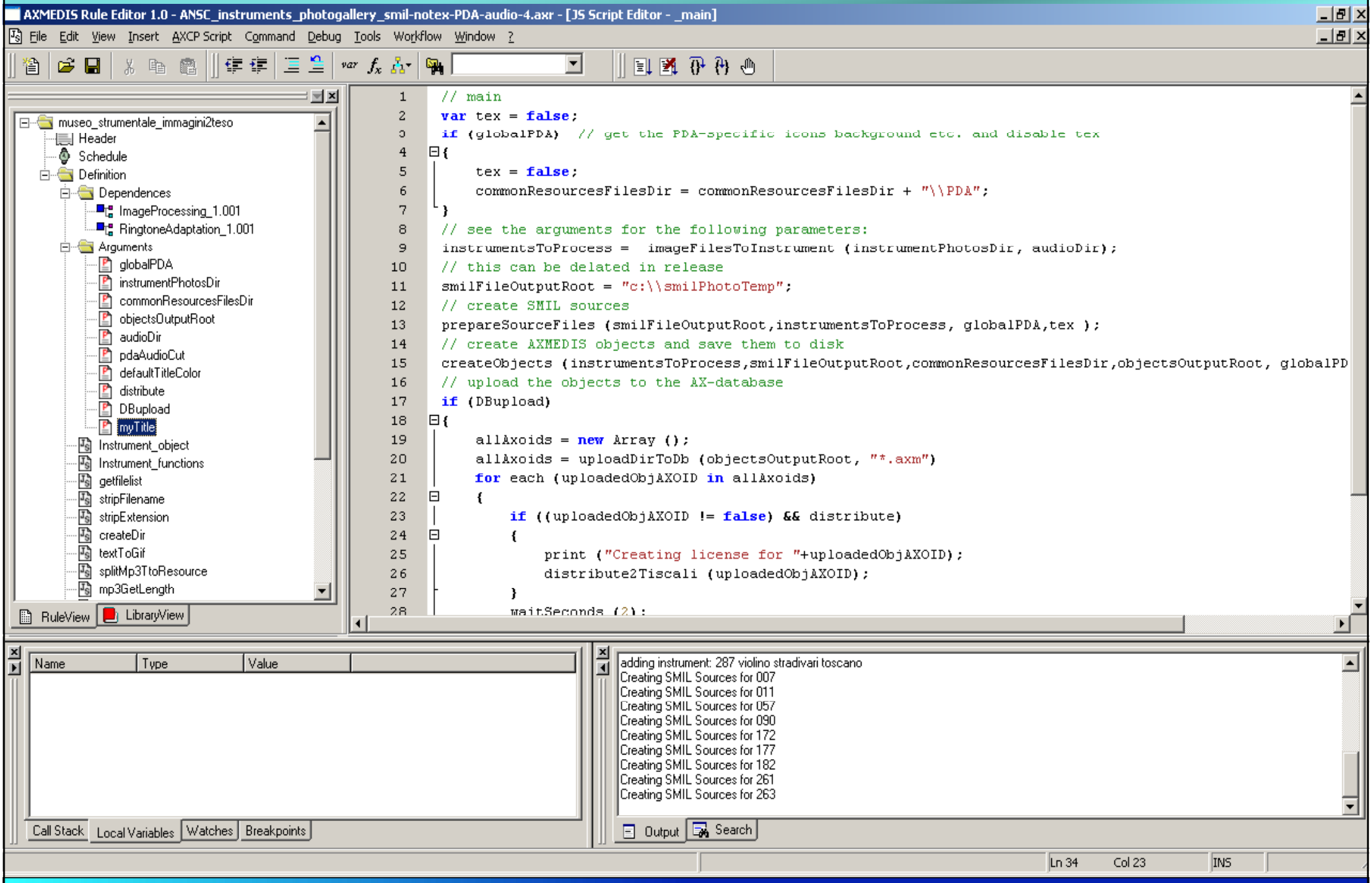
Main reqs. for visual media proc.

- **visual programming of**
  - single process flow on the single executing node and hierarchies;
- **visual processing of media, including Blocks/Functions:**
  - content processing, communication, semantic processing, taking decision engine, reasoning, transcoding, adaptation, etc.;
- **hierarchical structure of the processes**
  - a process may activate other processes on other Grids or on the same Grid;
  - Complex and branched flows composed by several different processes, allocated on different Grid nodes;
- **error code reporting**
  - in the single process and in the branched processes on the grid;
- **construction of single processing procedures/elements**
  - Storing/retrieving them on/from database,
  - allocation on the grid nodes according to scheduling strategies.

AXMEDIS Content Processing GRID



Snapshot from the AXCP IDE front page



Analysis of the status (>280 rules, several producers)

A. **75% single rules with linear structure,**

   a sequence of activities to be performed.

   - when one of the activity fails the whole rule execution has to fail.
   - To this category belong rules for automated content production on demand, licensing, content publication and/or repurposing, etc. Return sync or async.

B. **9% rules activated by other Rules on the Grid in asynchronous manner.**

   - mother rule does not need to wait for the result to continue its running. are structurally realized as rules of type A.
   - they start asynchronously and do not need to keep blocked the main rule.

C. **16% rules activating/invoking other processing Rules by creating synchronous/asynchronous derived Rules waiting/or-not for their completion to continue their execution.**

Analysis (Continue)

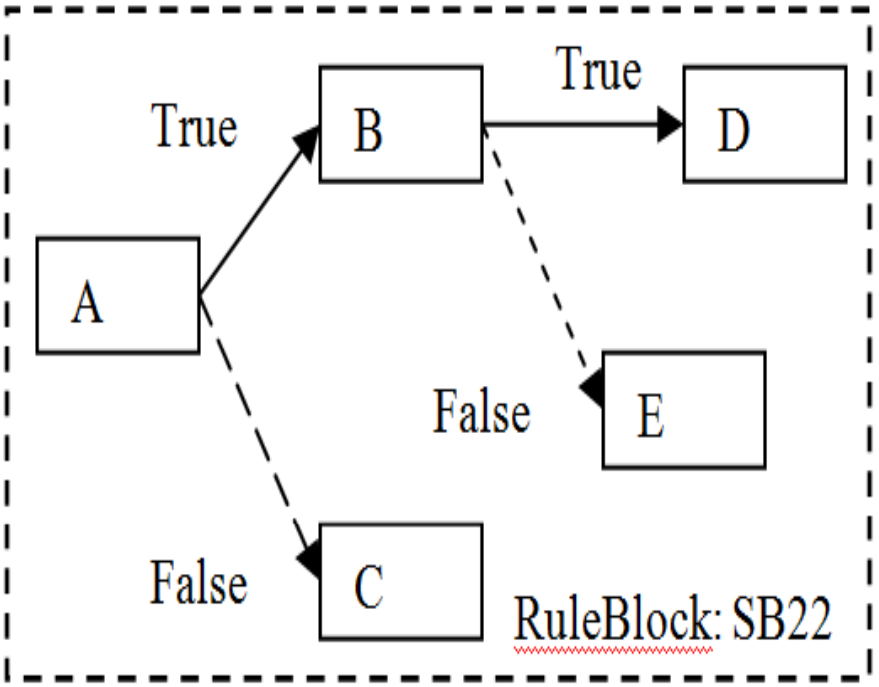
- **almost all rules present JS segments of functional blocks**
  - working on single or on lists of content elements
  - performing specific activities such as:
    - content ingestion, query actualization, content retrieval, storage, adaptation, extraction and processing descriptors, transcoding, synchronisation, fingerprint estimation, indexing, summarization, metadata manipulation, com..etc.
- **AXVD Visual Language allows to compose:**
  - single elements of the process (called JSBlock) to create composed Rules allocated on the same processor node (covering rule of type A and B)
  - branching activities (collection of RuleBlocks) which are allocated and executed on the Grid infrastructure according to their dependency by the scheduler. The Rules capable to activate other Rules cover the specific semantic of rules of type C, identified in the analysis.

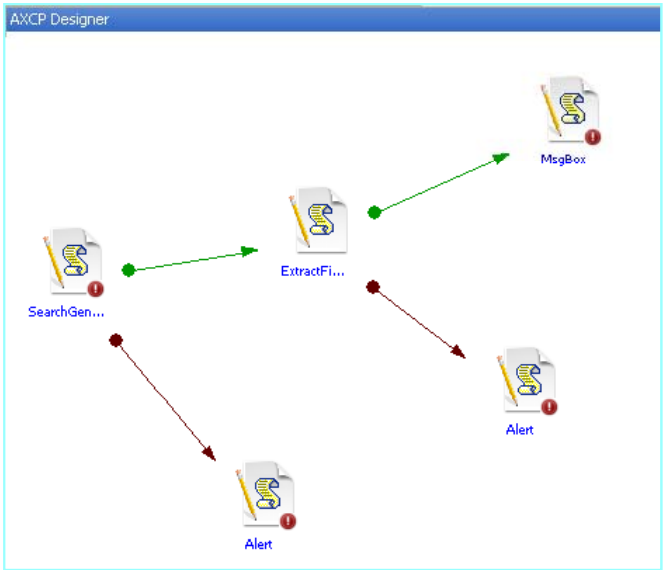
JSBlock

- **A JSBlock is characterized by a name (type name and instance) and a set of in/out parameters. A parameter can be marked as:**
  - IN when it is consumed into the Rule,
    - ➔ Types of elements works on list of elements reducing the needs for explicit iterations.
  - OUT when it is consumed into the rule and can be used to pass back a result to the next processing segments, *that is IN/OUT*,
  - SETUP (static) when it is a reserved INput to set up block specific behaviour. This parameter type is used to force different operating conditions in the Block. For example, to pass the ID of the database to be used, the temporary directory, etc.
- **Each JSBlock brings the processing flow towards one of the two possible directions of execution:**
  - (i) without errors, or
  - (ii) with error occurrence.

A RuleBlock from JSBlocks

- The scope of In/OUT parameters is passed along the hierarchy.
- Only one leaf is reached in the graph at the execution.
- The generator produces the code for the GRID and can put in execution the RuleBlock on debug
- Each JSBlock can activate other RuleBlocks as well.
- In the visual editor red/green coding is given to the arrows error/success completion.



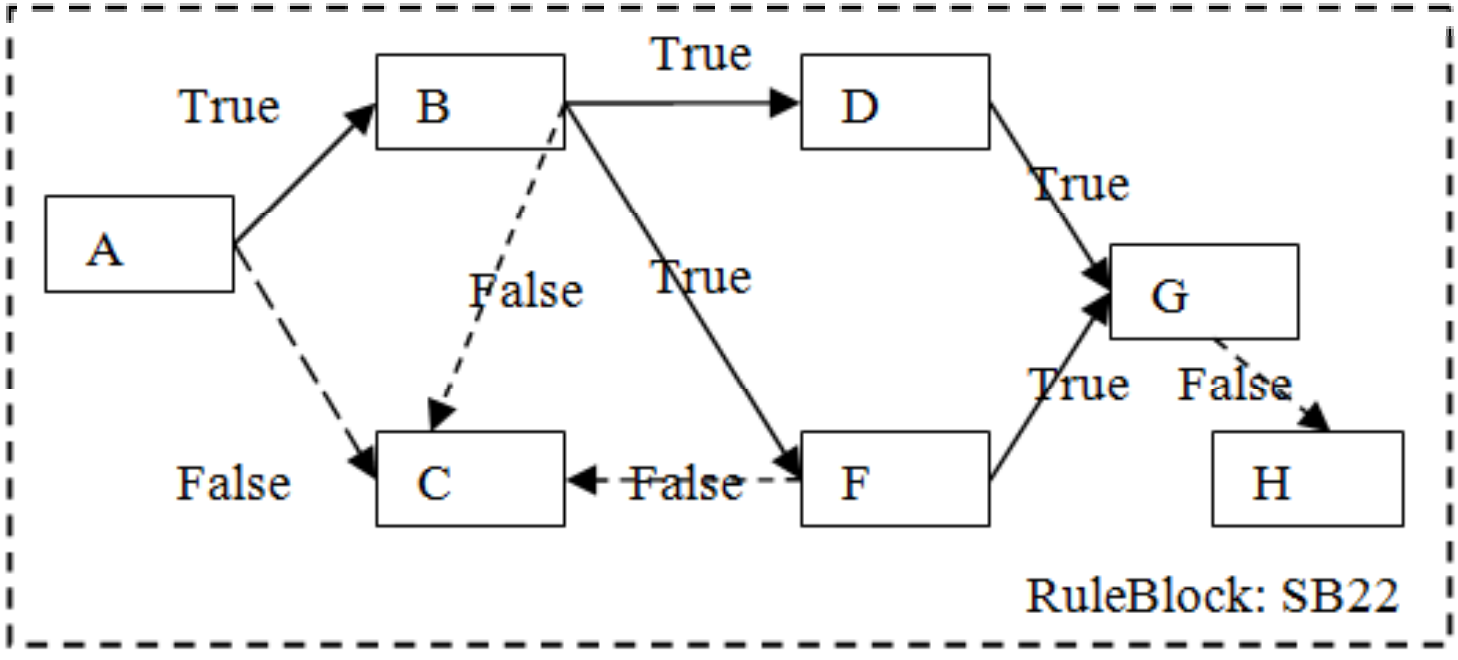


Iterations and decisions

- **iterations are internally managed into the single JSBlocks.**
  - The IN/OUT parameters have types based on collections and lists of items
  - For example, a JSBlock performing transcoding works indifferently receiving in input a single file reference or a list;
  - Conceptually, for the users, the single element is only a specific case of an array of them.
- **decisions can be taken into the single JSBlock.**
  - A JSBlock can be regarded as a visual implementation of a selection and/or of a sequence of actions.
  - A single JSBlock can be used to evaluate any kind of Boolean condition, which allows to take a decision.
  - Multiple decisions can be implemented as chains of JSBlocks as in the if-then-else-if–then-else construct.
  - Thus a JSBlock can implement complex firing conditions as one can do in Petri Nets.

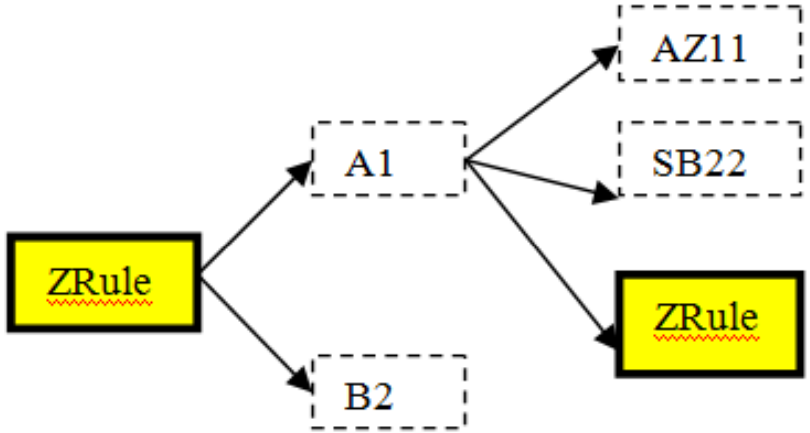
Extensions in place now

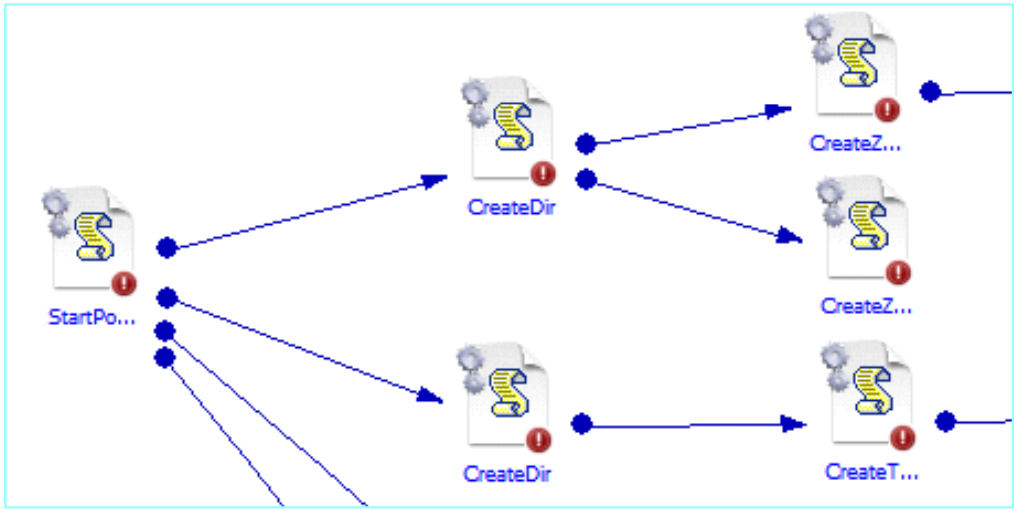
- **A new version of the language and model has been defined extending the described version with diamond connections among JSBlocks. This allows to**
  - access at a larger scope of variables coming from several blocks
  - avoid duplicating JSBlocks in the same diagram
  - Better manage the errors codes that could come from multiple JSBlocks as well and thus it could be used to realize RuleBlocks which could define recovering paths into the graph



Managing RuleBlocks Rules

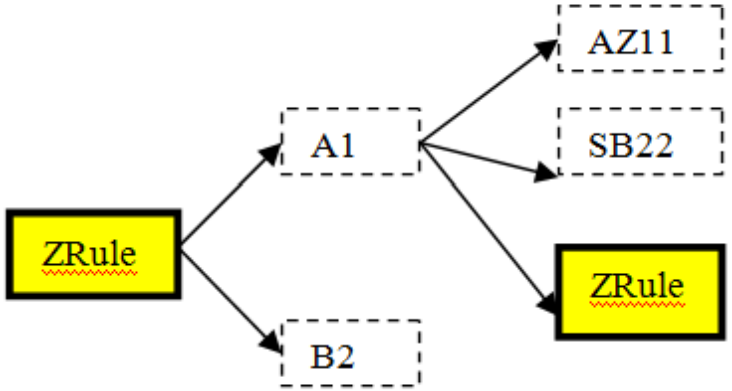
- **ManRuleBlocks are used to define the allocation and branching of Rules on grid nodes.**
- **A different visual semantics has been used**
  - Parallel and sequential execution of processes
  - child blocks as synchronous (blocked) or as asynchronous (non-blocked) processes
- **E.g.:**
  - Zrule is a ManRuleBlocks that presents recursive behavior.





Managing RuleBlocks Rules

- **IN/OUT parameter management and editing, a ManRuleBlock follows the same semantics of the RuleBlock**



- **The production implies**
  - The access to database to get the Block description, compose them and verify consistency
  - The errors are reported as results/OUT parameters. Thus they may be more complex that simple fault/nofault as in the JSBlocks

- **The generation and execution implies**
  - The production and allocation on the grid of a Managing Rule capable of organising the execution of those blocks on the whole GRID
  - The allocation of all the components rules and the interchange of their parameters on the GRID network.

Formal modeling of the language

- **A Visual Block is**
  - a segment of a JavaScript (JSBlock) or
  - a full RuleBlock (which in turn is created as a set of JSBlock or directly coded in JavaScript).

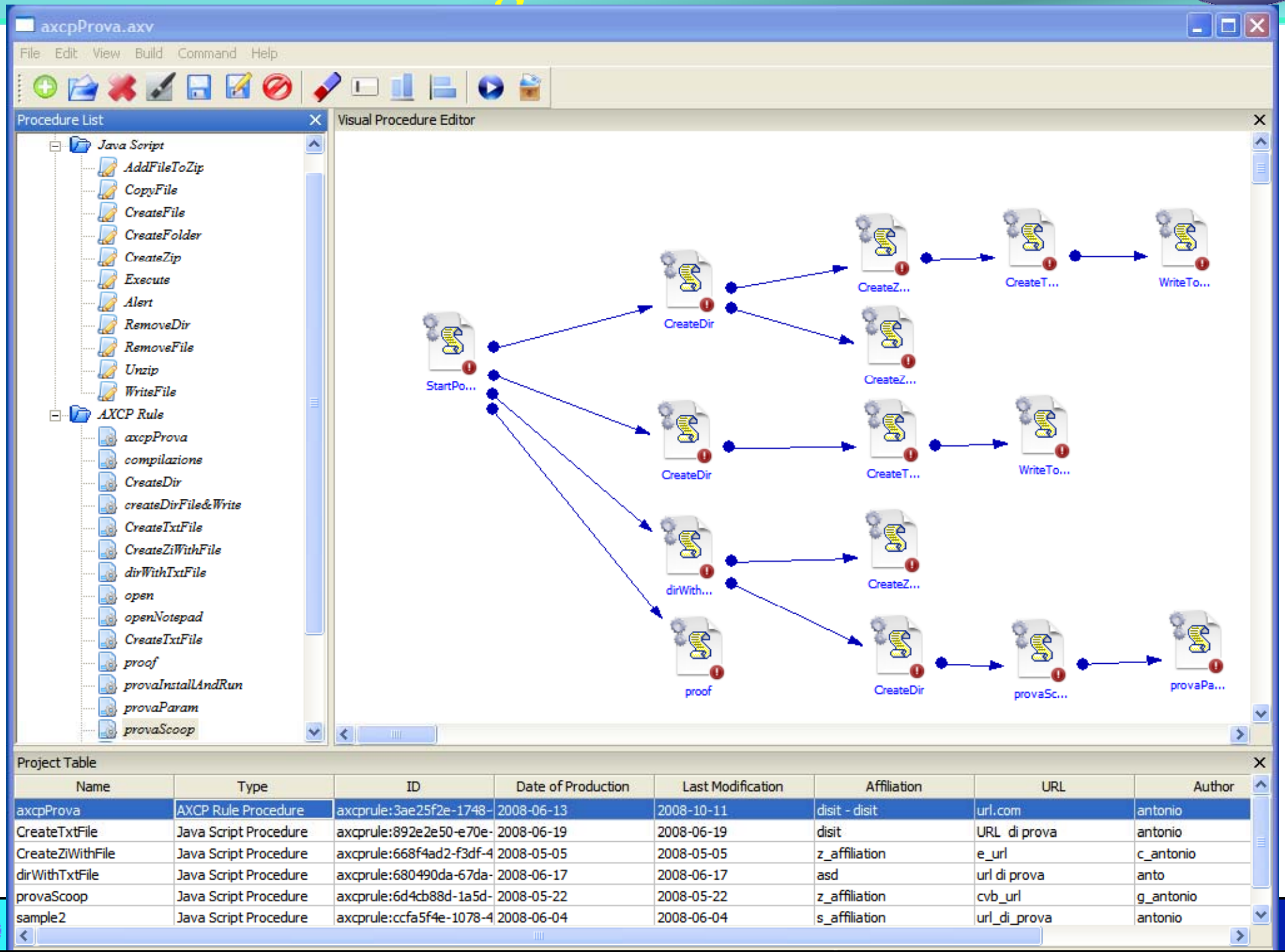
- **JSBlock::= <defined in JS as JS block>**

- **RuleBlock::= {JSBlock} | <defined in JS as JS rule>**

- **ManRuleBlock::= {RuleBlock | ManRuleBlock} |**
  <defined in JS as JS rule>

- This modeling allows to generate the code but also to manually produce it.

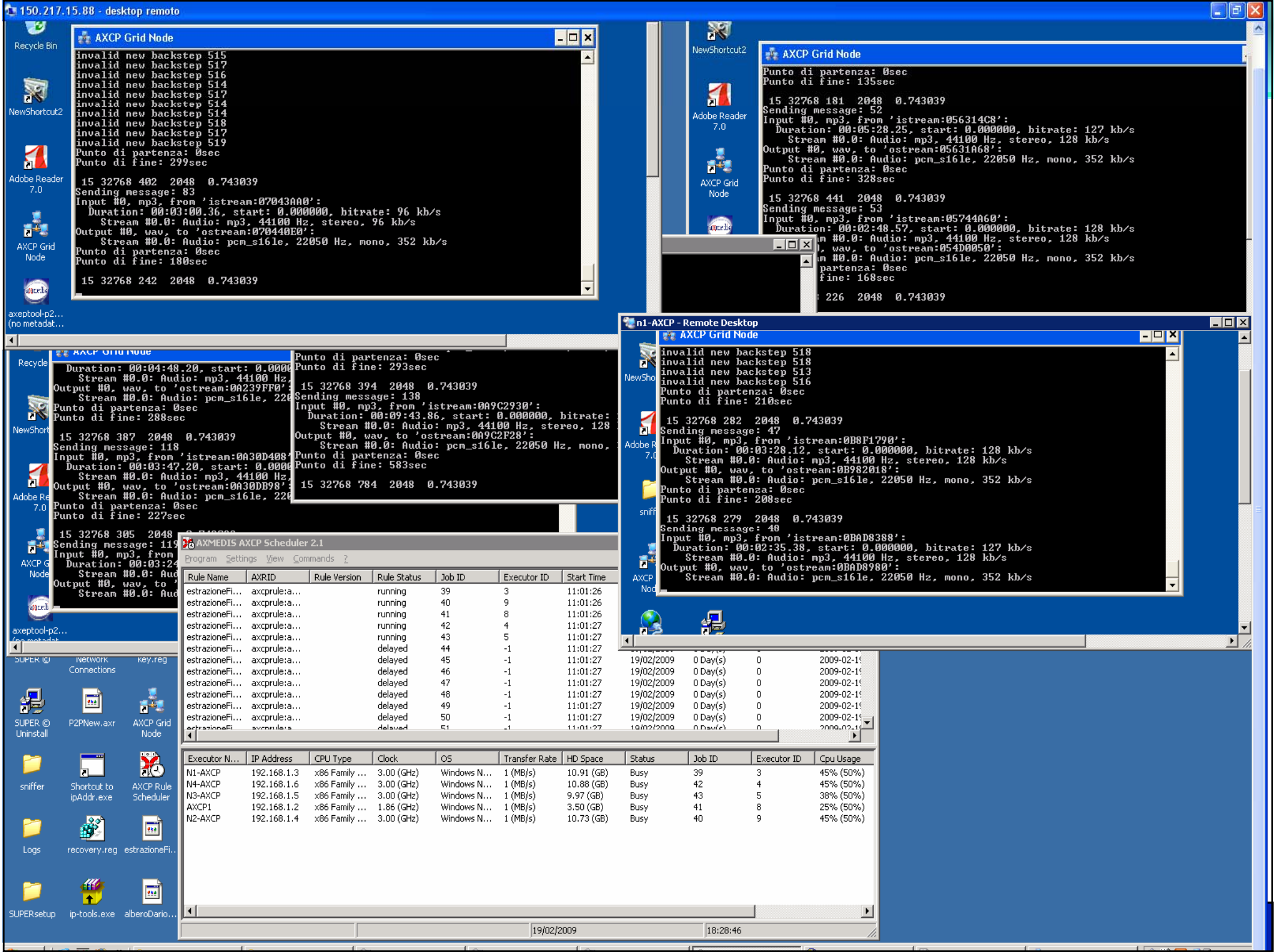
AXCP Visual Designer



Name	Type	ID	Date of Production	Last Modification	Affiliation	URL	Author
axcpProva	AXCP Rule Procedure	axcprule:3ae25f2e-1748-	2008-06-13	2008-10-11	disit - disit	url.com	antonio
CreateTxtFile	Java Script Procedure	axcprule:892e2e50-e70e-	2008-06-19	2008-06-19	disit	URL di prova	antonio
CreateZiWithFile	Java Script Procedure	axcprule:668f4ad2-f3df-4	2008-05-05	2008-05-05	z_affiliation	e_url	c_antonio
dirWithTxtFile	Java Script Procedure	axcprule:680490da-67da-	2008-06-17	2008-06-17	asd	url di prova	anto
provaScoop	Java Script Procedure	axcprule:6d4cb88d-1a5d-	2008-05-22	2008-05-22	z_affiliation	cvb_url	g_antonio
sample2	Java Script Procedure	axcprule:ccfa5f4e-1078-4	2008-06-04	2008-06-04	s_affiliation	url_di_prova	antonio

Some applicative examples

- **Applications are in the area of content processing**
  - Fingerprint extraction and collection for audio processing
  - Fingerprint recognition for audio track monitoring
  - P2P monitoring and network balance
  - P2P monitoring for audio/visual IPR monitoring and accounting
  - User generated content management according to different and dynamically changing kind of content and needs
  - Back office multichannel content production and distribution according to device and user profile, dynamic decisions have to be taken with GRID integrated taking decision systems
  - Etc.
- **It is a good instrument for**
  - highly dynamic back offices
  - Students development on media processing (stand alone node, single node, multiple nodes, hierarchical solutions).


150.217.15.88 - desktop remoto
AXCP Grid Node
invalid new backstep 515
invalid new backstep 517
invalid new backstep 516
invalid new backstep 514
invalid new backstep 517
invalid new backstep 514
invalid new backstep 514
invalid new backstep 518
invalid new backstep 517
invalid new backstep 519
Punto di partenza: 0sec
Punto di fine: 299sec
15 32768 402 2048 0.743039
Sending message: 83
Input #0, mp3, from 'istream:07043AA0':
Duration: 00:03:00.36, start: 0.000000, bitrate: 96 kb/s
Stream #0.0: Audio: mp3, 44100 Hz, stereo, 96 kb/s
Output #0, wav, to 'ostream:070440E0':
Stream #0.0: Audio: pcm_s16le, 22050 Hz, mono, 352 kb/s
Punto di partenza: 0sec
Punto di fine: 180sec
15 32768 242 2048 0.743039
AXCP Grid Node
Punto di partenza: 0sec
Punto di fine: 135sec
15 32768 181 2048 0.743039
Sending message: 52
Input #0, mp3, from 'istream:056314C8':
Duration: 00:05:28.25, start: 0.000000, bitrate: 127 kb/s
Stream #0.0: Audio: mp3, 44100 Hz, stereo, 128 kb/s
Output #0, wav, to 'ostream:05631A68':
Stream #0.0: Audio: pcm_s16le, 22050 Hz, mono, 352 kb/s
Punto di partenza: 0sec
Punto di fine: 328sec
15 32768 441 2048 0.743039
Sending message: 53
Input #0, mp3, from 'istream:05744A60':
Duration: 00:02:48.57, start: 0.000000, bitrate: 128 kb/s
Stream #0.0: Audio: mp3, 44100 Hz, stereo, 128 kb/s
Punto di partenza: 0sec
Punto di fine: 168sec
226 2048 0.743039
n1-AXCP - Remote Desktop
AXCP Grid Node
invalid new backstep 518
invalid new backstep 518
invalid new backstep 513
invalid new backstep 516
Punto di partenza: 0sec
Punto di fine: 210sec
15 32768 282 2048 0.743039
Sending message: 47
Input #0, mp3, from 'istream:0B8F1790':
Duration: 00:03:28.12, start: 0.000000, bitrate: 128 kb/s
Stream #0.0: Audio: mp3, 44100 Hz, stereo, 128 kb/s
Output #0, wav, to 'ostream:0B982018':
Stream #0.0: Audio: pcm_s16le, 22050 Hz, mono, 352 kb/s
Punto di partenza: 0sec
Punto di fine: 208sec
15 32768 279 2048 0.743039
Sending message: 48
Input #0, mp3, from 'istream:0BAD8388':
Duration: 00:02:35.38, start: 0.000000, bitrate: 127 kb/s
Stream #0.0: Audio: mp3, 44100 Hz, stereo, 128 kb/s
Output #0, wav, to 'ostream:0BAD8980':
Stream #0.0: Audio: pcm_s16le, 22050 Hz, mono, 352 kb/s
Punto di partenza: 0sec
Punto di fine: 293sec
15 32768 394 2048 0.743039
Sending message: 138
Input #0, mp3, from 'istream:0A9C2930':
Duration: 00:09:43.86, start: 0.000000, bitrate:
Stream #0.0: Audio: mp3, 44100 Hz, stereo, 128
Output #0, wav, to 'ostream:0A9C2F28':
Stream #0.0: Audio: pcm_s16le, 22050 Hz, mono,
Punto di partenza: 0sec
Punto di fine: 583sec
15 32768 784 2048 0.743039
Punto di partenza: 0sec
Punto di fine: 288sec
15 32768 387 2048 0.743039
Sending message: 118
Input #0, mp3, from 'istream:0A30D408'
Duration: 00:03:47.20, start: 0.0000
Stream #0.0: Audio: mp3, 44100 Hz,
Output #0, wav, to 'ostream:0A30DB98'
Stream #0.0: Audio: pcm_s16le, 220
Punto di partenza: 0sec
Punto di fine: 227sec
AXMEDIS AXCP Scheduler 2.1
Program Settings View Commands ?
Rule Name | AXRID | Rule Version | Rule Status | Job ID | Executor ID | Start Time
estrazioneFi... | axcprule:a... | | running | 39 | 3 | 11:01:26
estrazioneFi... | axcprule:a... | | running | 40 | 9 | 11:01:26
estrazioneFi... | axcprule:a... | | running | 41 | 8 | 11:01:26
estrazioneFi... | axcprule:a... | | running | 42 | 4 | 11:01:27
estrazioneFi... | axcprule:a... | | running | 43 | 5 | 11:01:27
estrazioneFi... | axcprule:a... | | delayed | 44 | -1 | 11:01:27
estrazioneFi... | axcprule:a... | | delayed | 45 | -1 | 11:01:27
estrazioneFi... | axcprule:a... | | delayed | 46 | -1 | 11:01:27
estrazioneFi... | axcprule:a... | | delayed | 47 | -1 | 11:01:27
estrazioneFi... | axcprule:a... | | delayed | 48 | -1 | 11:01:27
estrazioneFi... | axcprule:a... | | delayed | 49 | -1 | 11:01:27
estrazioneFi... | axcprule:a... | | delayed | 50 | -1 | 11:01:27
19/02/2009 | 0 Day(s) | 0 | 2009-02-19
Executor N... | IP Address | CPU Type | Clock | OS | Transfer Rate | HD Space | Status | Job ID | Executor ID | Cpu Usage
N1-AXCP | 192.168.1.3 | x86 Family ... | 3.00 (GHz) | Windows N... | 1 (MB/s) | 10.91 (GB) | Busy | 39 | 3 | 45% (50%)
N4-AXCP | 192.168.1.6 | x86 Family ... | 3.00 (GHz) | Windows N... | 1 (MB/s) | 10.88 (GB) | Busy | 42 | 4 | 45% (50%)
N3-AXCP | 192.168.1.5 | x86 Family ... | 3.00 (GHz) | Windows N... | 1 (MB/s) | 9.97 (GB) | Busy | 43 | 5 | 38% (50%)
AXCP1 | 192.168.1.2 | x86 Family ... | 1.86 (GHz) | Windows N... | 1 (MB/s) | 3.50 (GB) | Busy | 41 | 8 | 25% (50%)
N2-AXCP | 192.168.1.4 | x86 Family ... | 3.00 (GHz) | Windows N... | 1 (MB/s) | 10.73 (GB) | Busy | 40 | 9 | 45% (50%)
19/02/2009
18:28:46
Recycle Bin
NewShortcut2
Adobe Reader 7.0
AXCP Grid Node
axeptool-p2...
Network Connections
key.reg
SUPER © Uninstall
P2PNew.axr
sniffer
Shortcut to ipAddr.exe
AXCP Rule Scheduler
Logs
recovery.reg
estrazioneFi...
SUPERsetup
ip-tools.exe
alberoDario...

Conclusions

- **The Visual language for programming grid processing has been defined and validated against several applications**
- **A new version of the AXVD language and model has been also defined extending the described version with diamond connections among JSBlocks**
- **The tool realizing the AXVD is integrated into the AXCP IDE grid processing tools of AXMEDIS solution for content processing**
  - They are freely distributed as freeware and are accessible free of charge for any no-profit institutions.
  - An open source (dual licensing) version will be published in the next months for any no-profit institutions
- **AXCP It is presently in use on some projects, see AXMEDIS portals and on the web http://www.axmedis.org**