Domain Specific Languages, in practice

Federico Tomassetti
CEO at Strumenta

Domain Specific Languages… of all sizes and shapes

Internal vs External

Horizontal vs Vertical

Federico Tomassetti

STRU
MEN
TA

Internal DSLs: the NOT so interesting Domain Specific Languages

An internal DSL is just a particular idiom of writing code in the host language. So a Ruby internal DSL is Ruby code, just written in particular style which gives a more language-like feel. As such they are often called Fluent Interfaces or Embedded DSLs

Martin Fowler

Internal DSLs: the NOT so interesting Domain Specific Languages

```ruby
Rails.application.routes.draw do
  root to: "pages#main"

  resources :posts do
    get :preview

    resources :comments, only:
            [:new, :create, :destroy]
  end
end
```

Federico Tomassetti

STRU MEN TA

Internal DSLs: the NOT so interesting Domain Specific Languages

```
cursor = database.factory()
    .select(Files.ID,
            Files.BACKUP_PATH)
    .from(Files.FILES)
  .leftOuterJoin(Entries.ENTRIES)
    .on(Entries.FILE_ID.equal(Files.ID))
  .where(Entries.FILE_ID.isNull())
  .fetchLazy();
```

External DSLs: the really interesting Domain Specific Languages

An external DSL is a completely separate language that is parsed into data that the host language can understand

Martin Fowler

i.e., a real language, created to do few things, well.

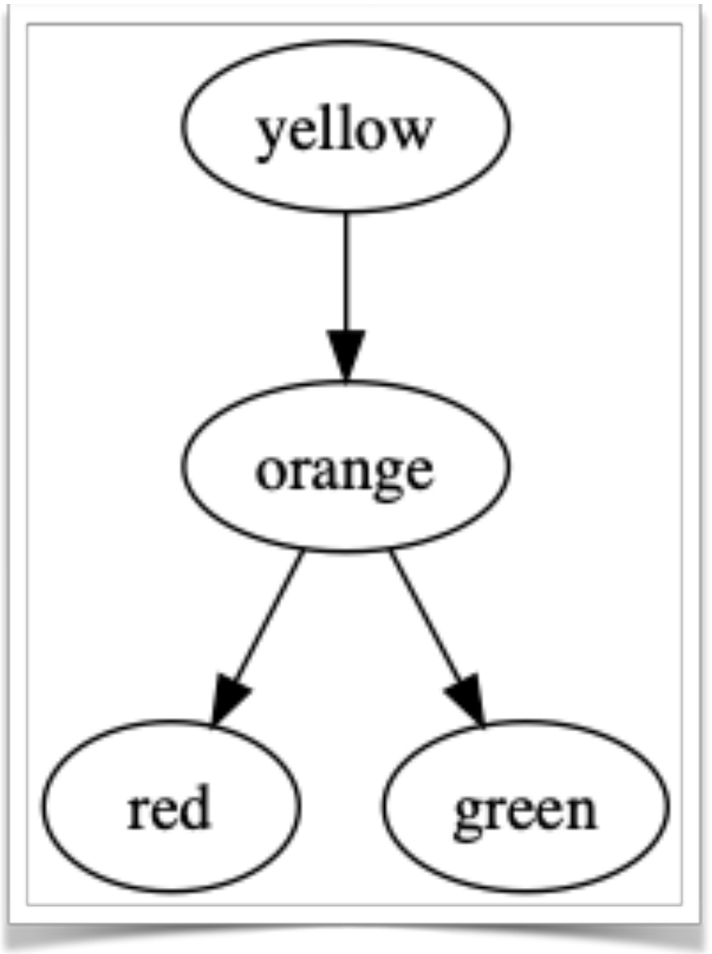
What external Domain Specific Languages are?

- Let’s see some examples of well known external DSLs
- Let’s reason on them
- Let’s dive on vertical DSLs built for specific organizations
- Let’s look into how we can build that stuff
- Considerations on DSLs

Examples of external DSLs

Dot, a small DSL to define graphs:

```
digraph graphname {
    yellow -> orange -> red;
    orange -> green;
}
```



Examples of external DSLs

SQL, a DSL to define queries:

```
SELECT MAX(TEMP_F), MIN(TEMP_F), AVG(RAIN_I), ID
FROM STATS
GROUP BY ID;
```

Examples of external DSLs

VHDL, a DSL to define circuits:

```
DFF : process(RST, CLK) is
begin
if RST = '1' then
Q <= '0';
elsif rising_edge(CLK) then
Q <= D;
end if;
end process DFF;
```

Examples of external DSLs

ANTLR, a DSL to define grammars:

```
// Identifiers
ID : [_]*[a-z][A-Za-z0-9_]* ;
// Literals
INTLIT : '0'|[1-9][0-9]* ;
DECLIT : '0'|[1-9][0-9]* '.' [0-9]+ ;
STRINGLIT : '"' ~["]* '"' ;
```

Some characteristics of external DSLs

- They reduce what can go wrong (NPE?)
- They can be used with limited training
- We can build tool support for them: auto-completion, syntax highlighting, error checking
- They are concise: a lot can be achieved changing a couple of lines
- They are portable: today we can generate a PNG diagram, tomorrow draw the diagram using SVG. Or move to another DB

Domain Specific Languages… of all sizes and shapes

Internal vs External

Horizontal vs Vertical

Federico Tomassetti

STRU
MEN
TA

Domain Specific Languages… of all sizes and shapes

Let’s see what sort of DSLs can be built for companies

Disclaimer: I can only show limited information that are made publicly available

DSL for Medical Software

```
decision tree [redacted]DecisionTree(score: [redacted]OverBaseline,
                                     patientHasAnySymptom: boolean, goToStartBrat: boolean)
patientHasAnySymptom
    score >= 7
        score in [4..6]
            goToStartBrat
                [redacted] RecoSBrat
                [redacted] RecoCBrat
    [redacted] Reco1
    [redacted] Reco3
    [redacted] Reco2
```

Federico Tomassetti

STRU
MEN
TA

DSL for Medical Software

```
decision table BpScoreDecisionTable(sys: bpRange, dia: bpRange) =
```

		dia					
		<= 50	[51..90]	[91..95]	[96..100]	[101..109]	>= 110
sys	<= 90	1	1	3	4	5	6
	[91..140]	2	2	3	4	5	6
	[141..150]	3	3	3	4	5	6
	[151..160]	4	4	4	4	5	6
	[161..179]	5	5	5	5	5	6
	>= 180	6	6	6	6	6	6

DSL for Medical Software

```
scenario scenario_8
  global timeout:    1 hours
  time granularity: 60 seconds

given
  inputPainBaseline          = 1
  inputPainMedicineDuration = Six

when
  at 0 min: EventInPainMeasure answers to PainMeasureQuestionnaire {
                                  measure: 4
                                }
            EventInPainSymptoms1 answers to PainSymptoms1Questionnaire {
                                  interferingDailyActivities: false
                                  newSite                   : false
                                  interferingAbilityToWalk  : false
                                }
then
  at 0 hours: assert parent sent message Recommendation(PainRecoSymptom1, Six)
  at 29 min:  assert parent in state     PainMeasure.Ask
  at 30 min:  assert parent in state     PainInitial.Ask
```

DSL for Medical Software

Federico Tomassetti

STRUMENTA

DSL for Medical Software

What processes support?

- Development of medical applications, running on multiple targets
- Running all tests, in the IDE and on the mobile
- Coverage analysis
- Generation of documentation
- Integration with TFS

DSL for Medical Software

Two results I want to share:

1. Zero defects in the business logic several months after releasing the software.
2. Development time released by the typical 18 months to a few weeks

DSL for Public Administration

Stay tuned

unfortunately I cannot share anything specific at this time

Federico Tomassetti

STRU
MEN
TA

Dutch Tax Office


```
regel aantal dagen ZVW-plicht 02        geldig vanaf dd. 01-01-2013        bron: Regeling Zorgverzekering, art 5.9; Zorgverzekeringswet, art 2

het aantal dagen ZVW-plicht van een IB-plichtige moet gesteld worden op de maximale waarde van A en B
indien hij aan alle volgende voorwaarden voldoet:
- zijn begindatum premieplicht ZVW is gevuld
- zijn datum ingang actief-militair is gevuld
- zijn datum beeindiging actief-militair is gevuld.
Daarbij geldt:
  A is (maand uit zijn einddatum premieplicht ZVW min maand uit zijn begindatum premieplicht ZVW) maal
       het AANTAL DAGEN IN MAAND plus (dag uit zijn einddatum premieplicht ZVW min
       dag uit zijn begindatum premieplicht ZVW) min (
       maand uit zijn datum beeindiging actief-militair min maand uit zijn datum ingang actief-militair) maal
       het AANTAL DAGEN IN MAAND plus (dag uit zijn datum beeindiging actief-militair min
       dag uit zijn datum ingang actief-militair)
  B is 0.
```

243 billion

€ tax revenues

11.2 million

tax returns received

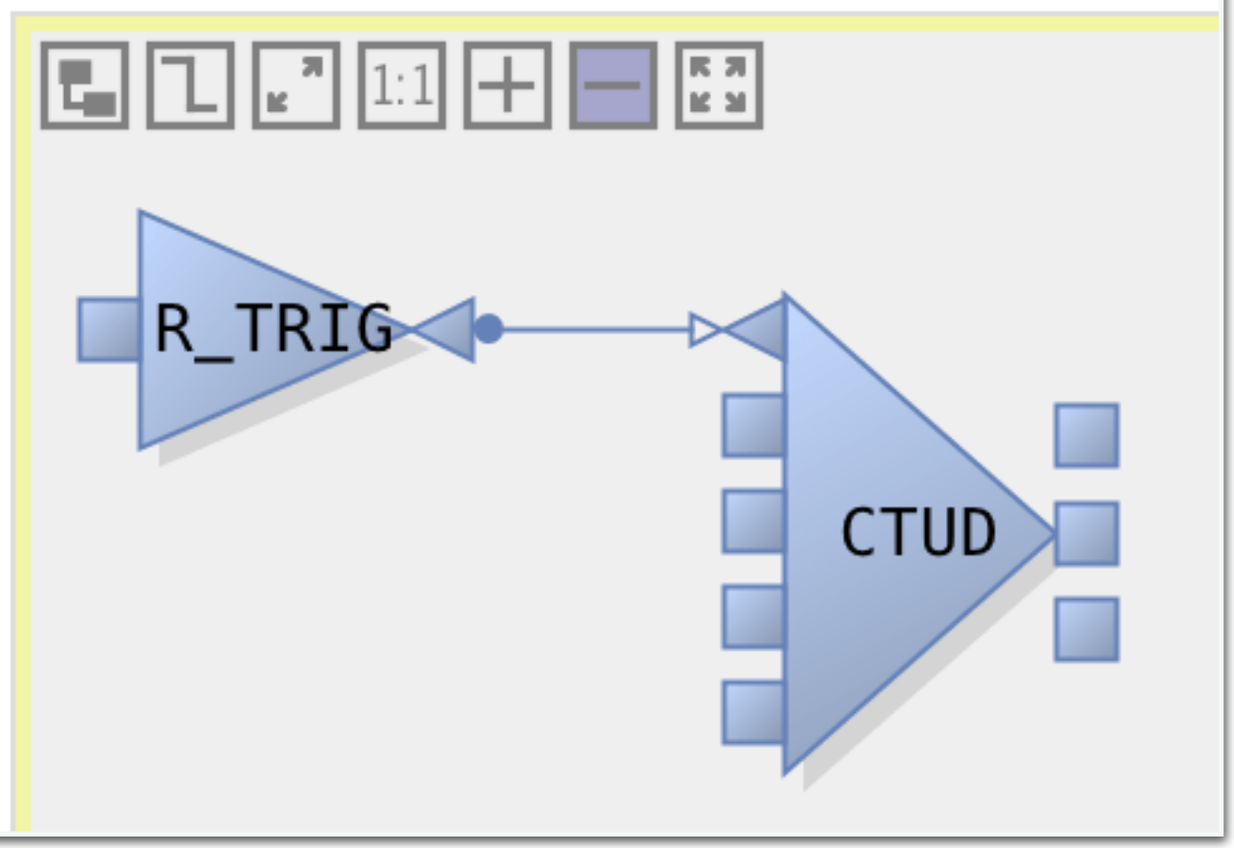
Federico Tomassetti

STRU MEN TA

Hardella IDE

```
PROGRAM CFC_DEMO
  variables:
    boilerEnabled : BOOL;
    temperature : INT;

  body:
    CFC
```



Smart programming environment for PLC

https://hardella.com/en/

Federico Tomassetti

STRUMENTA

Characteristics of vertical DSLs

- Cover all sort of processes
- Have different roles using different DSLs
- Several DSLs are integrated
- The effort in building supporting tools is serious (months/years of work)
- They change how an organization work
- The results can be really impressive (20 times faster development, 0 bugs)

How do we implement real DSLs?

Three technologies I suggest:

- **ANTLR**: if you need extra flexibility for integration with specific toolchains

- **Xtext**: if your DSL target developers

- **Jetbrains MPS**: in most cases, specifically if you want to create powerful and complete solutions for domain experts who are not developers

What is MPS?

MPS is a Language Workbench

MPS is based on projectional editing

MPS is developed by Jetbrains

Formats DSL

It is open-source.

It has been built as a demo, it is not production ready.

https://github.com/Strumenta/formatsdsl

Federico Tomassetti

STRU
MEN
TA

Formats DSL

4.1. The `ClassFile` Structure

A `class` file consists of a single `ClassFile` structure:

```
ClassFile {
    u4             magic;
    u2             minor_version;
    u2             major_version;
    u2             constant_pool_count;
    cp_info        constant_pool[constant_pool_count-1];
    u2             access_flags;
    u2             this_class;
    u2             super_class;
    u2             interfaces_count;
    u2             interfaces[interfaces_count];
    u2             fields_count;
    field_info     fields[fields_count];
    u2             methods_count;
    method_info    methods[methods_count];
    u2             attributes_count;
    attribute_info attributes[attributes_count];
}
```

Formats DSL

```
cp_info {
    u1 tag;
    u1 info[];
}
```

Each item in the `constant_pool` table must begin with a 1-byte tag indicating the kind of `cp_info` entry. The contents of the `info` array vary with the value of `tag`. The valid tags and their values are listed in Table 4.3. Each tag byte must be followed by two or more bytes giving information about the specific constant. The format of the additional information varies with the tag value.

Table 4.3. Constant pool tags

Constant Type	Value
`CONSTANT_Class`	7
`CONSTANT_Fieldref`	9
`CONSTANT_Methodref`	10
`CONSTANT_InterfaceMethodref`	11
`CONSTANT_String`	8
`CONSTANT_Integer`	3
`CONSTANT_Float`	4

Formats DSL

3.2. Chunk layout

Each chunk consists of four parts:

Length

A 4-byte unsigned integer giving the number of bytes in the chunk's data field. The len
decoders should treat the length as unsigned, its value must not exceed 2^{31} bytes.

Chunk Type

A 4-byte chunk type code. For convenience in description and in examining PNG files
However, encoders and decoders must treat the codes as fixed binary values, not chara
Additional naming conventions for chunk types are discussed in the next section.

Chunk Data

The data bytes appropriate to the chunk type, if any. This field can be of zero length.

CRC

A 4-byte CRC (Cyclic Redundancy Check) calculated on the preceding bytes in the ch
chunks containing no data. See CRC algorithm.

Formats DSL

The IHDR chunk must appear FIRST. It contains:

```
Width:                  4 bytes
Height:                 4 bytes
Bit depth:              1 byte
Color type:             1 byte
Compression method:     1 byte
Filter method:          1 byte
Interlace method:       1 byte
```

Width and height give the image dimensions in pixels. They

How much this stuff cost?

It depends…

Federico Tomassetti

STRU
MEN
TA

How much this stuff cost?

Simple DSL with editors and interpreter:

- Prototype: 2 man-weeks
- Production ready: 2 man-months

Advanced DSL with editors, interpreter, documentation generator, simulators, testing support

- Prototype: 2-4 man-months
- Production ready: 2-5 man-years

Benefits

Development speed: because we have a language tailored at what we do, which is concise and abstract the technological details away

Code quality: the language makes most errors simply not possible. We can also get high-level error checking and specific testing support. We also have much less code

Platform independence: we capture knowledge in a format that is technology independent. We can move to a different platform by rewriting the interpreter or code generator, without touching none of our knowledge.

Who benefit the most from DSL?

- Product companies: DSLs are ideal if you build a lot of similar stuff
- Companies who build software but their core competence is something else: e.g., companies building medical software or accounting software
- Companies with domain experts: engineers, medical doctors, accountants, financial experts, etc.

How to introduce DSLs in a company?

DSLs are build to support how people work and permit to improve existing processes through better tool support.

It is natural to get resistance from some of the stakeholders, while getting a lot of support from others.

Typically developers resist them and domain experts are enthusiastic.

So you need to communicate well and involve developers as early as possible.

Final considerations

This stuff is not a tiny fluent interface. It is something real: big benefits, significative investments needed.

Real DSLs are core assets we use to capture knowledge. Once we capture knowledge we can build processes around it.

Around processes we build supporting tools.

They change how organizations work make them orders of magnitude more effective.

Federico Tomassetti

STRU
MEN
TA

Final considerations

Building DSLs is about one thing:

Giving better tools to people to do their job.

People have skills and expertise. If we give them good tools they multiply what they can achieve.

Learn more about DSLs
(and get some free resources)

https://tomassetti.me/working-software/

Federico Tomassetti

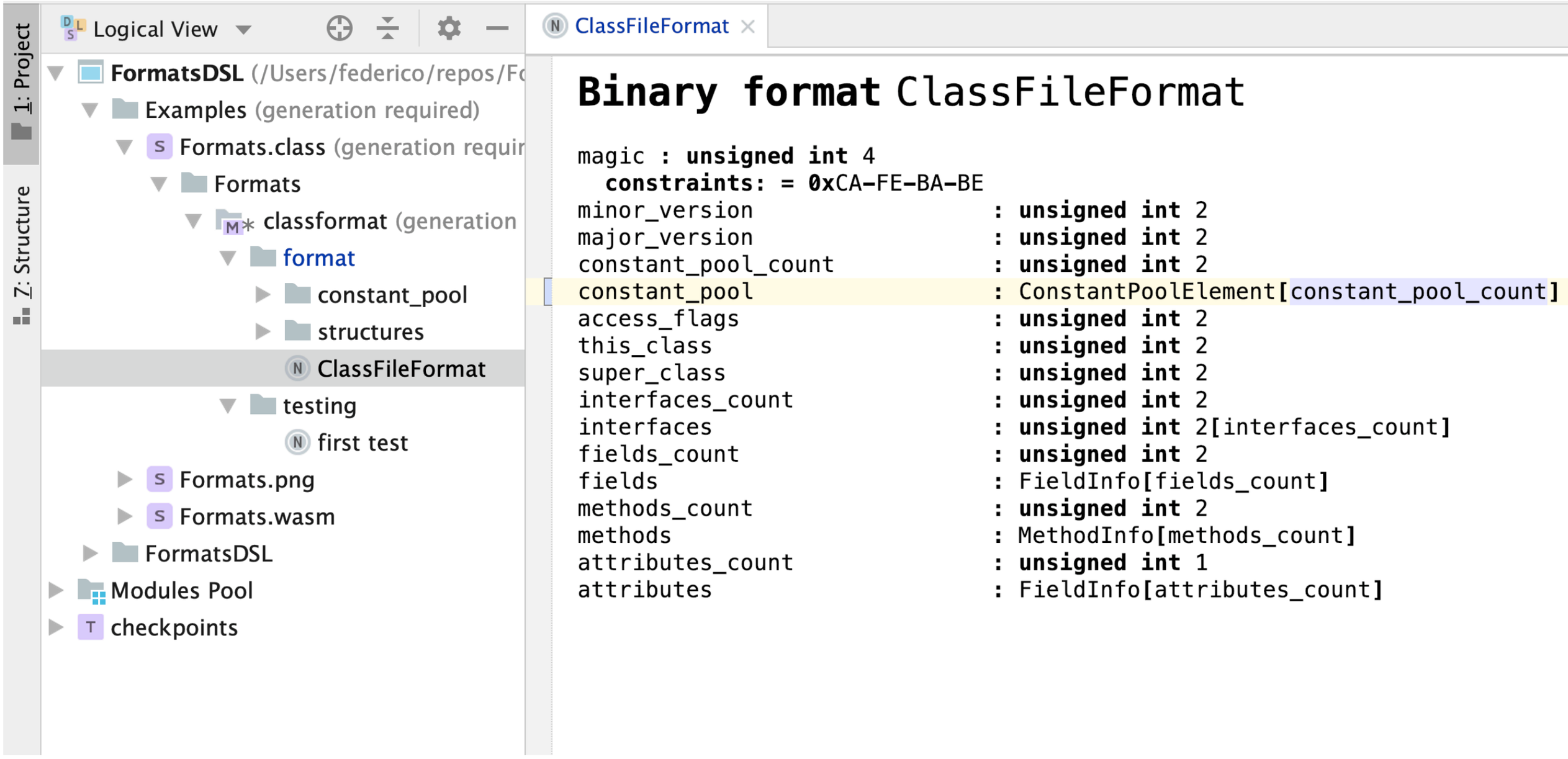
STRU
MEN
TA

DSLs & Agile

DSLs are to me the ultimate agile practice because

- They make the change concise
- They make the change obvious
- The change can be performed and executed by more stakeholders

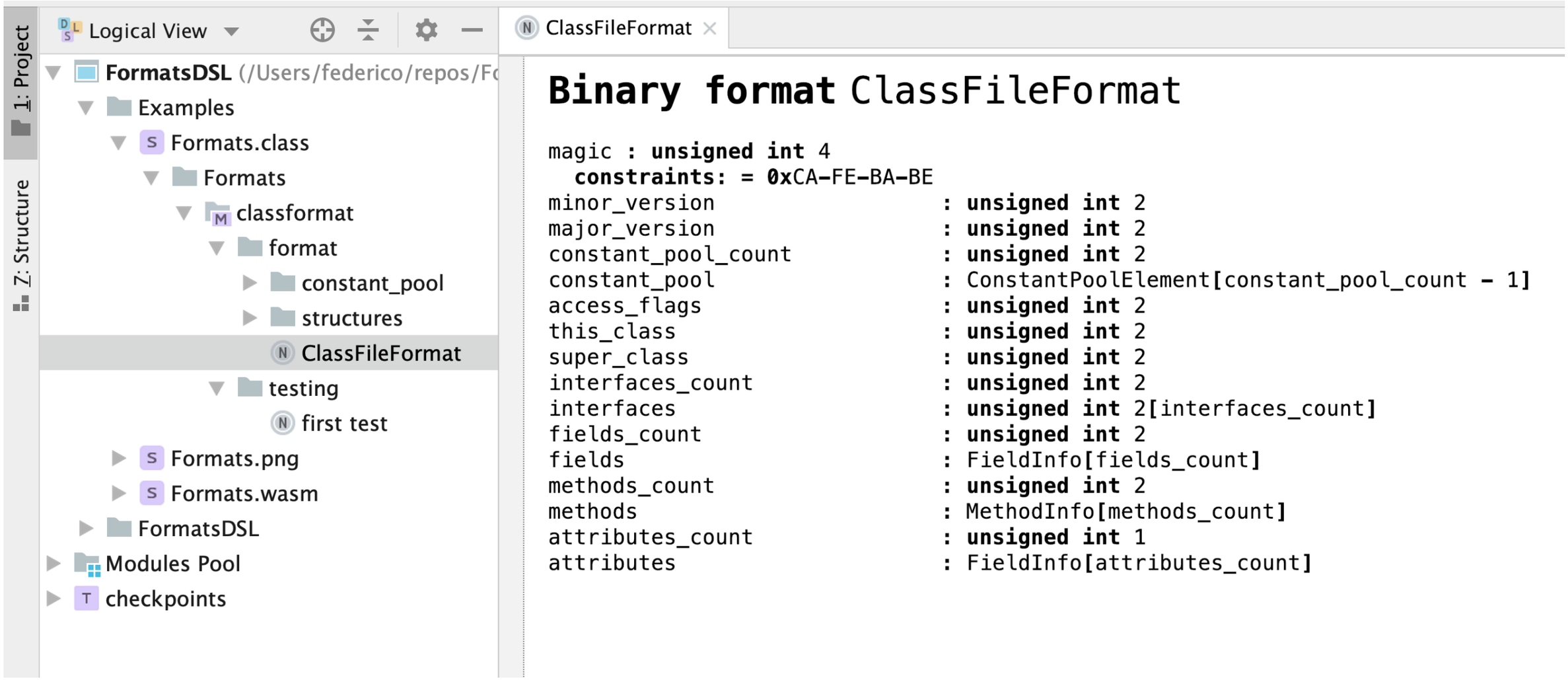
DSLs & Agile



Federico Tomassetti

STRU
MEN
TA

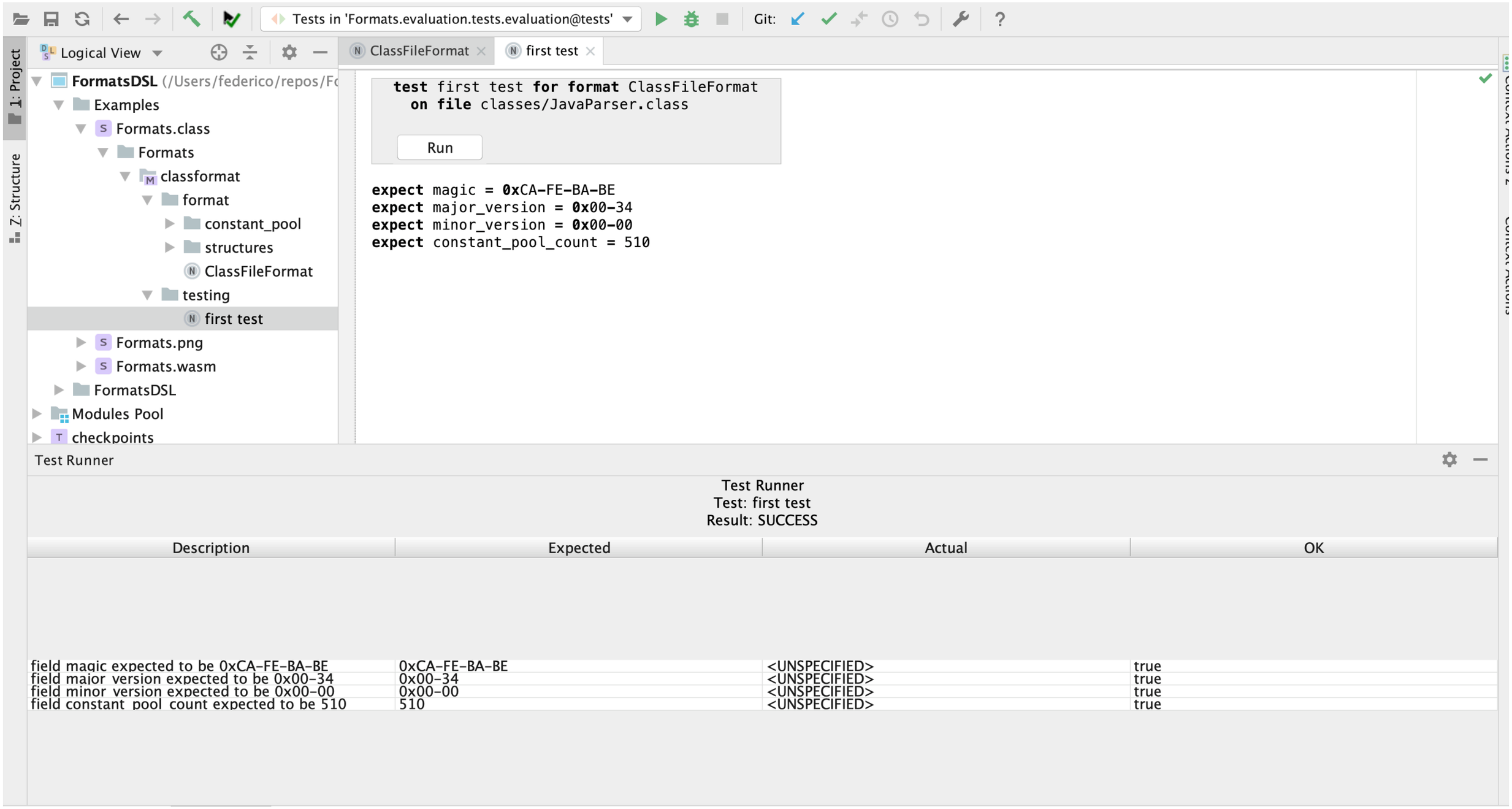
DSLs & Agile



Federico Tomassetti

STRUMENTA

DSLs & Agile



Federico Tomassetti

STRUMENTA