



# ALCHEMIST

Smart Document & Source Code Auditor

22 June 2017

W.T. (Wim) Goes | Directeur Valori Software Improvement  
**VALORI** | Orteliuslaan 1000 | UTRECHT | **NL**  
T +31 30 711 11 11 | F +31 30 711 11 12 | M +31 6 5 0 99 9 66 8  
E [WIMGOES@VALORI.NL](mailto:WIMGOES@VALORI.NL) | [WWW.VALORI.NL](http://WWW.VALORI.NL)

# Contents

---

1. Introduction
2. Functional overview
3. Platform overview
4. Domain and data models
5. Document domain

# 1. Alchemist – Introduction

---

- Alchemist an intelligent text mining and assessment tool with added value in different domains. Currently used for software source code analysis
  - Software quality monitoring for documentation and source code
  - Software reconstruction and documentation
- The unique characteristics of Alchemist are
  - Combination of powerful text mining technologies
  - Flexible text mining language, including software specific functions
  - High scalability, text input and output
  - Small footprint, on premise usage
  - Based on the lightweight, high performance language Lua

## 2. Alchemist – Functional Overview (1)

### Added Value

Reports  
(progress & quality)

Insight  
(business & development)

Enhancement  
(maintenance)

Transformation  
(Models, VDML)



Source Code

Models

Documents

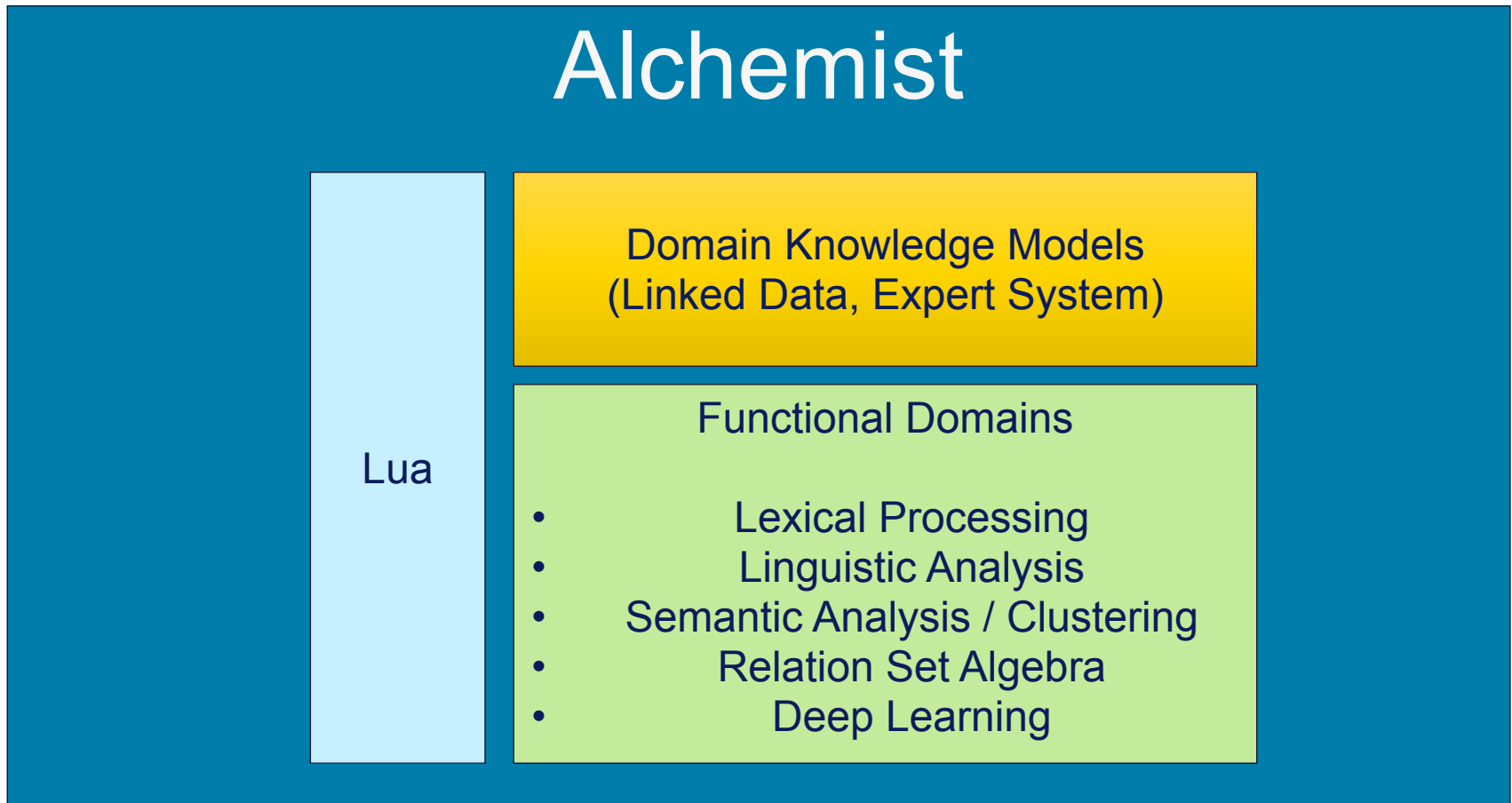
Logs

Testware

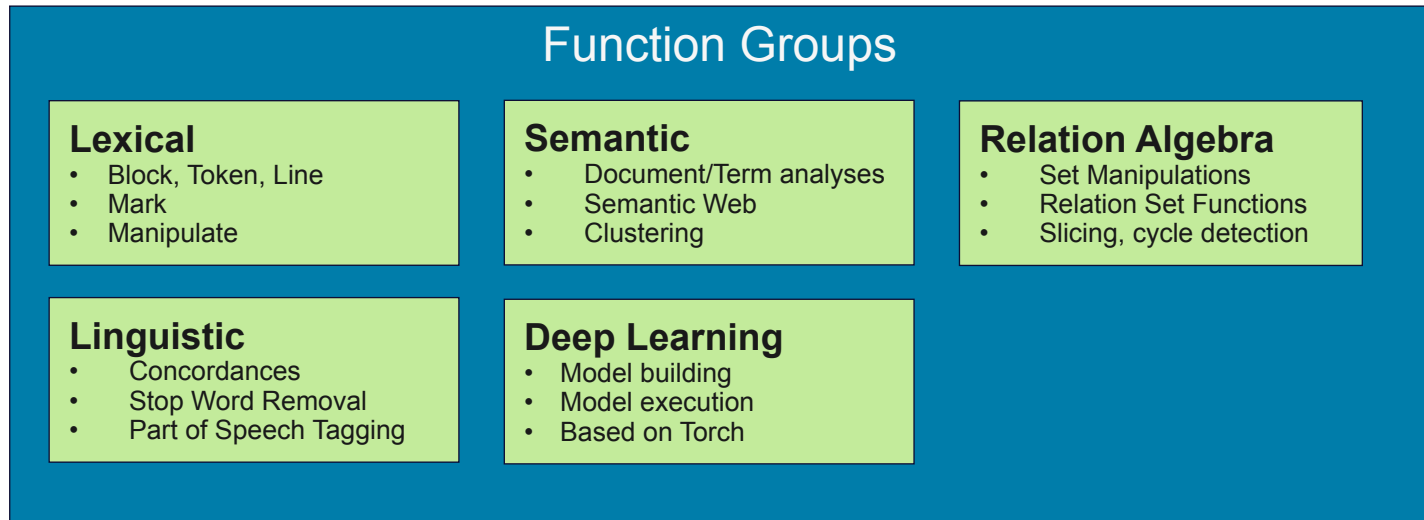
Administrations

### Artifacts

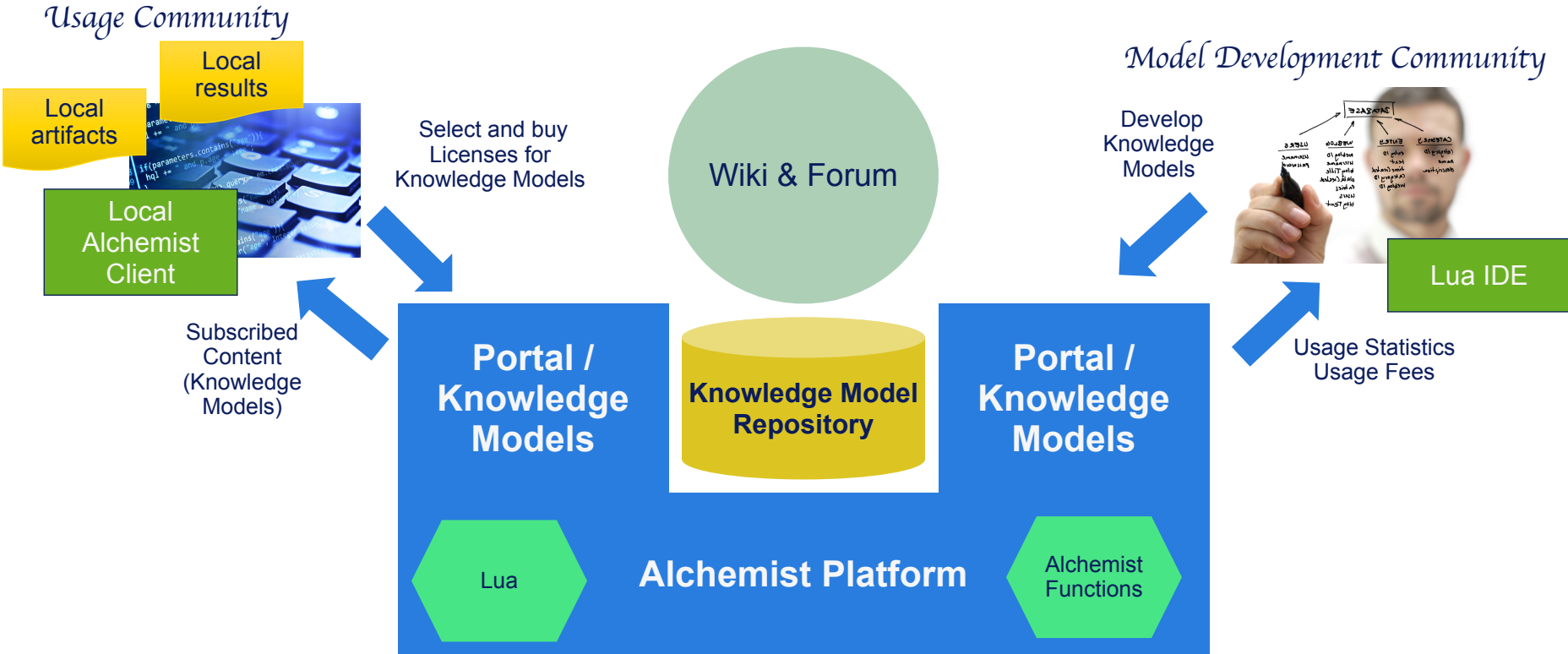
## 2. Alchemist – Functional Overview (2)



## 2. Alchemist - Functional Overview (3)



# 3. Alchemist - Platform Overview



## 4. Alchemist - Domain and Data Knowledge Models

---



Which method has the largest size?



Which methods are there in classes?



Which classes are there in Java files?



Which files are Java files?



Where are the files?

## 4. Alchemist - Domain and Data Knowledge Models



Which method has the largest size?



*Rules (Lua)*



Which methods are there in classes?



*Rules (Lua)*



Which classes are there in Java files?



*Rules (Lua)*



Which files are Java files?



*Rules (Lua)*



Where are the files?

# 4. Alchemist - Domain and Data Knowledge Models



Which method has the largest size?



*Rules (Lua)*

Which methods are there in classes?



*Rules (Lua)*

Which classes are there in Java files?



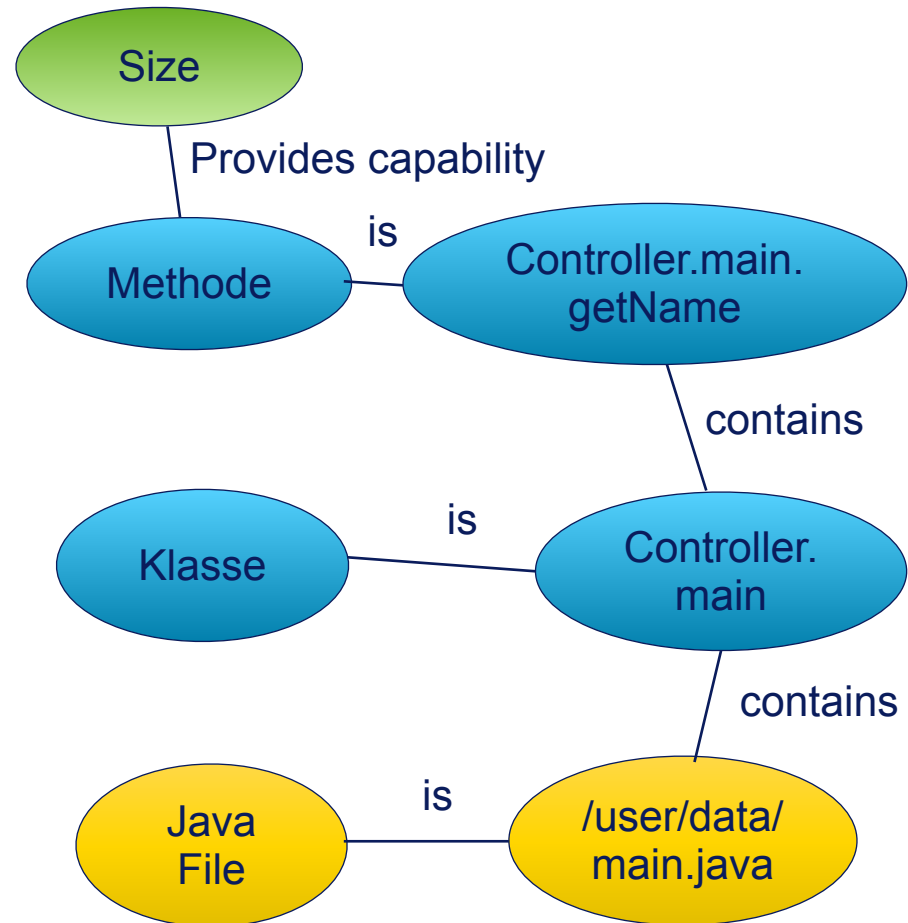
*Rules (Lua)*

Which files are Java files?

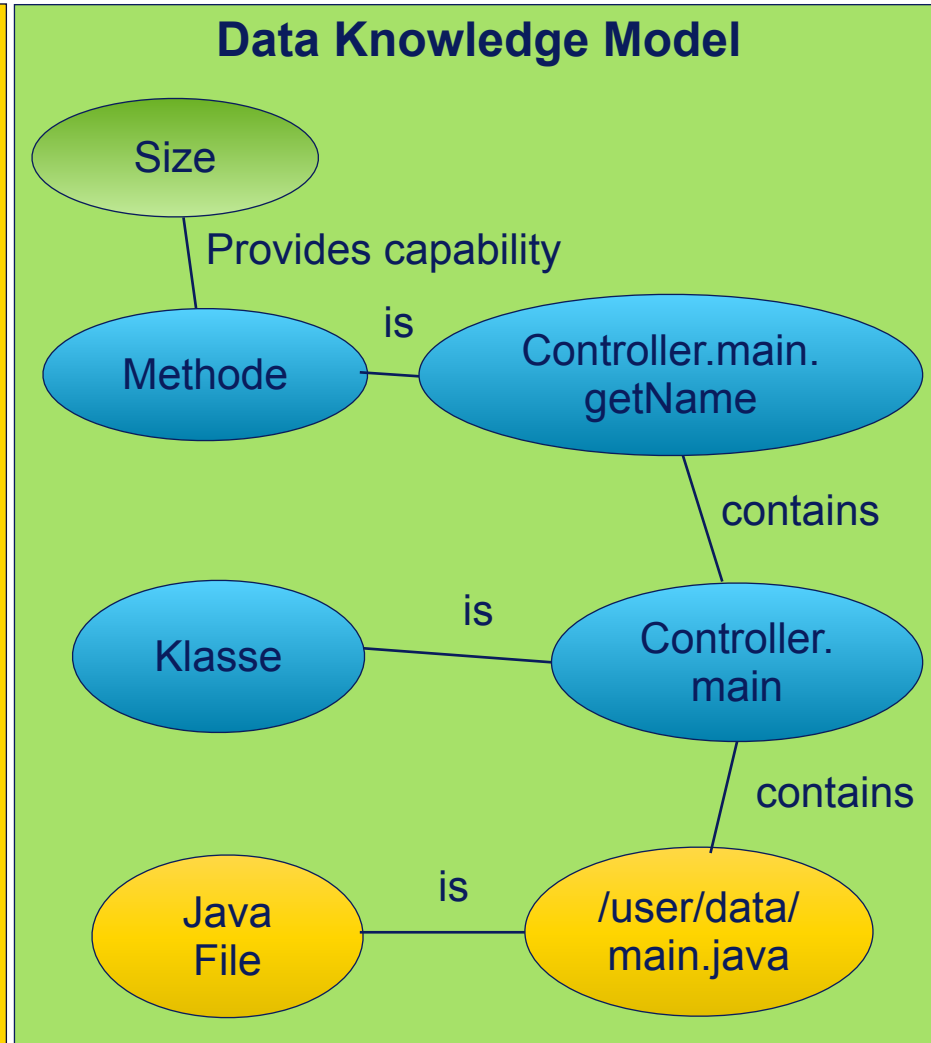
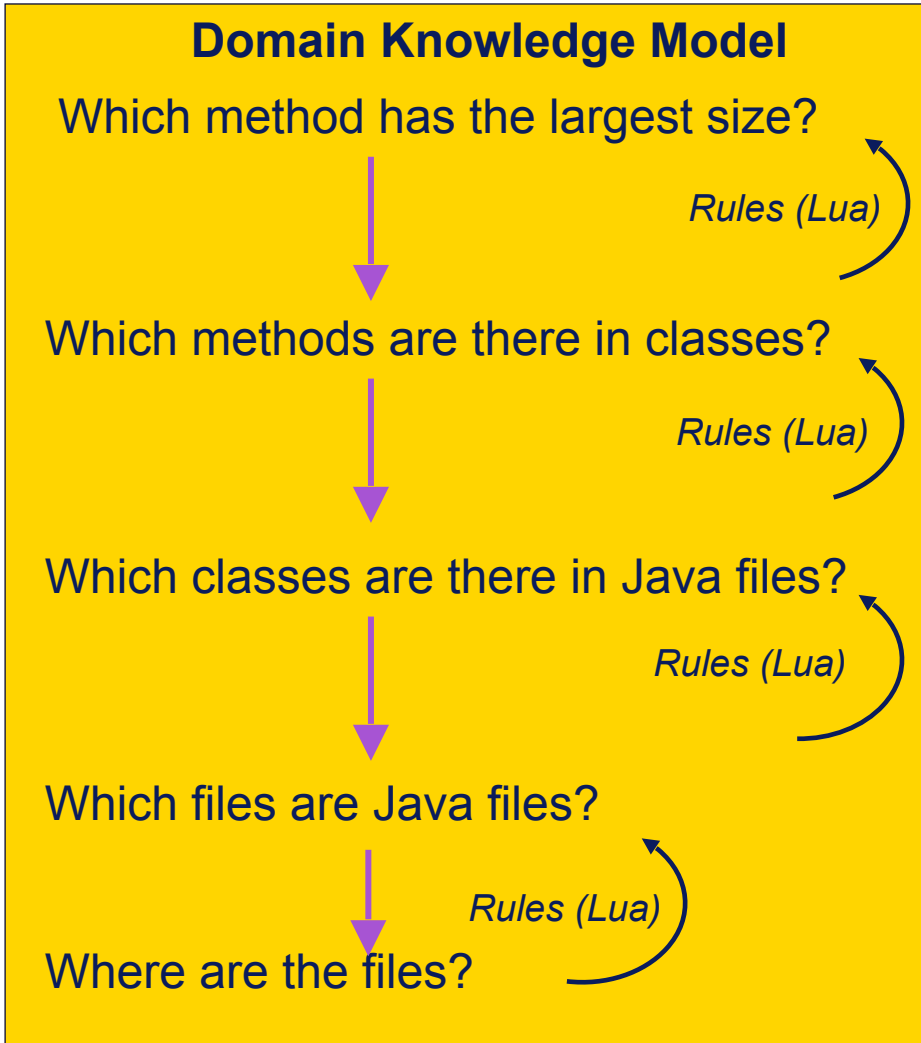


*Rules (Lua)*

Where are the files?



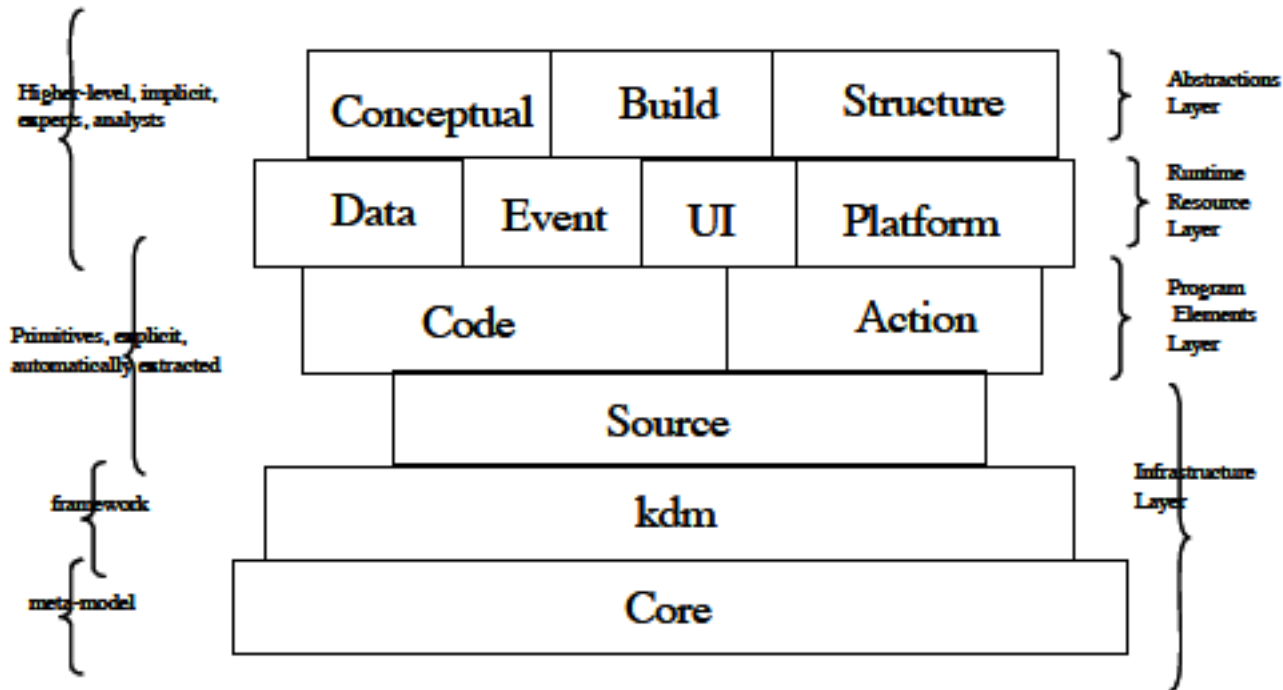
# 4. Alchemist - Domain and Data Knowledge Models



# 4. Alchemist - Domain and Data Knowledge Models



- Object Management Group (OMG) is a semantic model for representing source code at increasing levels of abstraction



# 4. Alchemist - Domain and Data Knowledge Models Lua implementation

```
KDM_Inventory_Java.domainKnowledge = {  
  {"createKDM_InventoryItem_Java", "creates", "KDM_InventoryItem_Java"},  
  {"createKDM_InventoryItem_Java", "creates", "KDM_InventoryItem_Text"},  
  {  
    name="createKDM_InventoryItem_Java",  
    selects = { { "inventoryItems", "?", "is", "KDM_InventoryItem" } },  
    action = function (self) KDM_Inventory_Java.create(self.inventoryItems, self.controller) end  
  }  
}
```

```
{ "createKDM_Metric_Codelines_Java", "providesCapability", "KDM_Metric_codelines"},  
{  
  name="createKDM_Metric_Codelines_Java",  
  condition = function(self) return self.controller:select({self.item.name, "is", "KDM_Method_Unit_Java"}) end,  
  capability = function(self) return KDM_Method_Unit_Java.getCodelines(self) end  
},
```

```
{ "createKDM_Metric_McCabe_Java", "providesCapability", "KDM_Metric_McCabe"},  
{  
  name="createKDM_Metric_McCabe_Java",  
  condition = function(self) return self.controller:select({self.item.name, "is", "KDM_Method_Unit_Java"}) end,  
  capability = function(self) return KDM_Method_Unit_Java.getMcCabe(self) end  
},
```

# 5. Alchemist – Handling documents

## -- Verdere discussie --

