The Evolutionary Approach to Semantics-Driven CBD Automation

Yu Jia

Institute of Software, Chinese Academy of Science

Beijing 100080, China

jia_yu@263.net

Two Problems in CBD Methodology

CBD Process:

- Multi-phrases:
  - Retrieve, qualification,adaptation,composition,deployment
- Different views to each phrase.Various techniques in each phase.

Problem 1: CBD paradigm lacks a commonly accepted principle to harmonize its engineering activities.

Component Semantics:

- IDLs are originally targeted for composition and communication.
- documents, test cases…. to be used as semantic carriers.

Problem 2: The component semantics is presented informal and independent from the component itself.

A Interesting Question

- EAs are search and optimization processes founded on self-organized individuals.
- the CBD is a software building procedure based on self-contained and highly independent components.

Is EA theory appropriate for CBD paradigm?

CBD vs. EA: The Corresponding Properties and Behaviors

- Self-organization
- Self-adaptation
- Self-learning
- Survival of the fittest
- Process
- Multi-solution and population search
- Statistics and random
- Intrinsic Parallelism
- Global optimization

CBD/EA theory

- The component is regarded as the evolvable individuals. The CBD process as the evolutionary procedure. So the algorithms in EA can be used as the automating tools in CBD.
- The Feature Modeling method is applied to describe the content and structure of component semantics.
  - Genotype: The Feature Description Logics is used to code the component
  - Phenotype: The meaning of Feature in a specific domain
- The user req. in Feature Space is used to calculate the fitness.
- All component-process activities can be categorized as 3 Operations: reproduction, crossover, and mutation.
- The object is to find or produce the nearly best solution in the in individual space: The target component-based system.

The Core of CBD/EA Approach

- A component model:
  Evolutionary component
- A engineering method:
  Feature Modelling
- A formal language:
  Feature Description Language
- A automation process:
  The evolutionary algorithms for CBD

The Evolvable Component Model

- Evolvable components are software entities implemented in CBD framework techniques and encoded in EA theory.
- Coming from the 3C model:

  component = (*CONCEPT*, *CONTENT*, *CONTEXT*)

  CONCEPT = (*SEM_INTERFACE*, *SYN_INTERFACE*)

  CONTEXT = (*SEM_ CONFIG*, *SYN_CONFIG*,*DOMAIN*)
- Considering only the semantic parts:

semantics = (*DOMAIN*, *SEM_INTERFACE*, *SEM_CONFIG*)

- Implemented in Feature Space:

$$semantics = (\Omega_{dom}, \Omega_{def}, \Omega_{con})$$

Feature Modeling

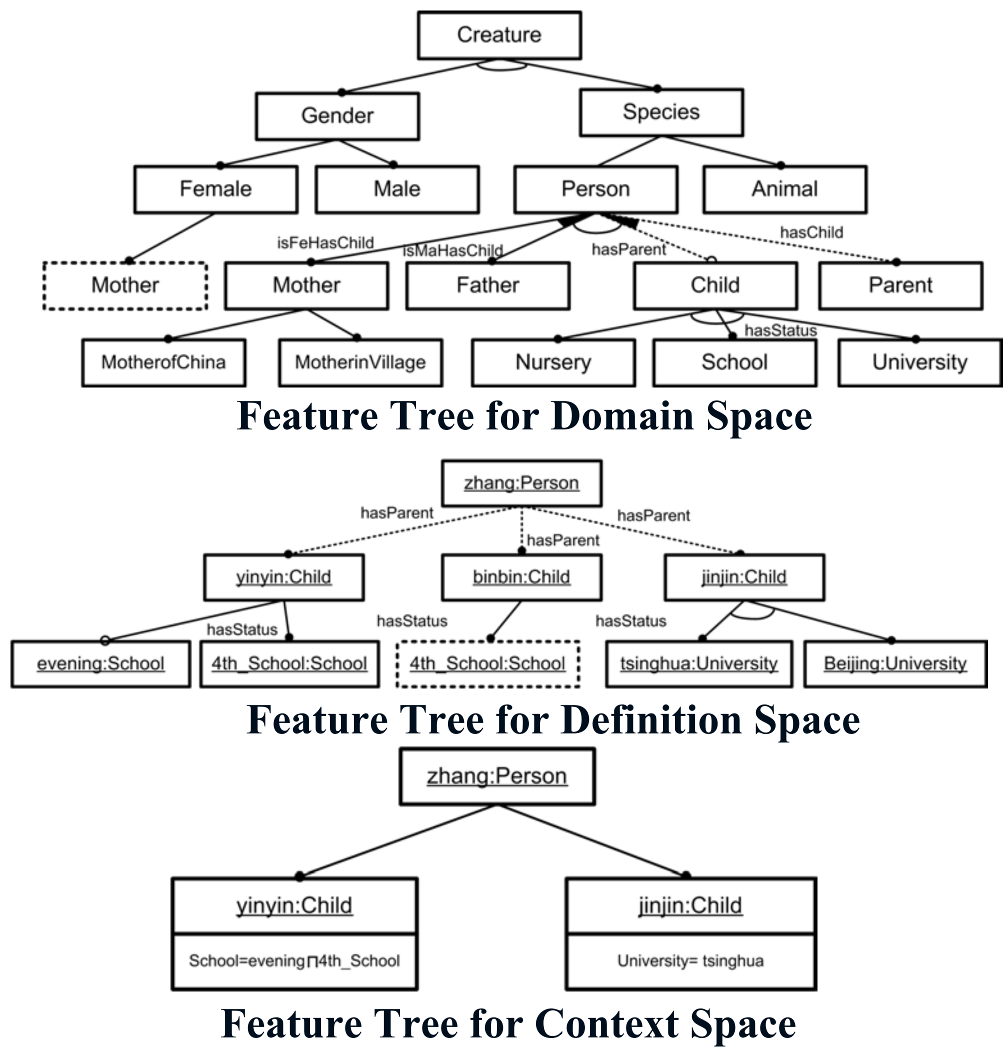
- "Feature" refers to concepts in a domain, mostly the services terms in perspective of user.
- A engineering methods in Separation of Concerns. The ontology theory in AI is its base.
- In CBD/EA, it used as:
  - A domain modelling engineering method;
  - A component semantics describing unit;
  - A unit to define user requirements;
  - The tools for analyzing or predicting the attributes of compositional system;
  - A base to design component evolution procedure and algorithms.

Book: K. Czarnecki, Generative Programming: Methods, Techniques, and Applications

Feature Modeling Techniques in CBD/EA

- The feature space:
  - **Domain Space, Definition Space, Context Space, and Requirement Space**
- The feature type:
  - **Mandatory, optional, alternative, and OR feature**
- The visual describing language: the Feature tree.
- A formal describing language: the feature description language.
- A set of feature model analyzing and verifying methods.
- Data structure and computing algorithms for feature space.
- A feature-oriented engineering framework:
  - The life cycle, A model specification, A feature dictionary Stakeholder's task,…

Feature tree: an example



Feature Description Logics (FDL)

- *Description Logics* (DLs) are knowledge representation languages for expressing knowledge about concepts and concept hierarchies.
- In DLs the domain of interest is modeled by means of *individuals*, *concepts*, *roles* and *knowledge base* exactly corresponding to the *feature items*, *features*, *feature relationships* and *Feature Space* respectively.
- The role of FDL in CBD/EA:
  - **A formal description language for component semantics;**
  - **A knowledge-based on the reuse method;**
  - **A intelligent tool to support CBD automation;**
  - **A genetic coding for evolutionary component**

The Syntax and semantics of FDL concept and role constructs

Features (Concepts) C	Syntax	Semantics
atomic feature	A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
universal feature	$\top$	$\Delta^{\mathcal{I}}$
negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
conjunction	$C_1 \sqcup C_2$	$C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}}$
alternation	$C_1 \vee C_2$	$(C_1^{\mathcal{I}} \cap \neg C_2^{\mathcal{I}}) \cup (\neg C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}})$
universal role quantification	$\forall R.C$	$\{o \mid \forall o' : (o, o') \in R^{\mathcal{I}} \rightarrow o' \in C^{\mathcal{I}}\}$
quantified number restriction	$(\leq n\ P.C)$	$\{o \mid \#\{o' \mid (o, o') \in P^{\mathcal{I}} \wedge o' \in C^{\mathcal{I}}\} \leq n\}$
collection of individuals	$\{a_1, \ldots, a_2\}$	$\{a_1^{\mathcal{I}}, \ldots, a_2^{\mathcal{I}}\}$
Feature Relations (Roles) R	**Syntax**	**Semantics**
atomic feature relations	P	$P^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
disjunction	$R_1 \sqcup R_2$	$R_1^{\mathcal{I}} \cup R_2^{\mathcal{I}}$
reverse	R^{-}	$\{(o, o') \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (o', o) \in R^{\mathcal{I}}\}$
concatenation	$R_1 \circ R_2$	$R_1^{\mathcal{I}} \circ R_2^{\mathcal{I}}$
reflect transitive closure	R^*	$(R^{\mathcal{I}})^*$

The Component Semantics in FDL

Notation Names	Expressions in $\mathcal{FDL}$	Examples
Feature Item	$C(a), D(b_1), D(b_2)$	Person (zhang), Child(yinyin)
Item Relation	$R(a, b)$	hasChild(zhang,yinyin)
Composite item	$C(a), D(b_1), D(b_2)$ $R(a, b_1)$, $R(a, b_2)$ or $C(a) \equiv R.D(b_1) \sqcup R.D(b_1)$	Person (zhang) Child(yinyin), Child(jinjin) hasChild(zhang,yinyin) hasChild(zhang,jinjin)
Control Item	$C(a)$ $\forall R^{-}.\bot(a)$	Person (zhang)
Basic Item	$C(a)$ $\forall R.\bot(a)$	University(tsinghua)
Shared Item	$C(a_1), C(a_2), D(b)$ $R(a_1, b), R(a_2, b)$	hasStatus(yinyin, 4th_School) hasStatus(binbin, 4th_School)
Alternative Relation	$C(a), D(b_1), D(b_2)$ $C(a) \equiv D(b_1) \vee D(b_2)$	Child(jinjin),University(tsinghua), University(Beijing) Child(jinjin)≡University(tsinghua), ∨University(Beijing)
Mandatory Relation	$C(a), D(b), R(a, b)$	Person (zhang), Child(yinyin) hasChild(zhang,yinyin)
Optional Relation	$C(a), D(b)$ $C(a) \sqsubseteq (\le 1 R.D(b))$	Child(yinyin)， School(evening) Child(yinyin)⊑(≤1 hasStatus.(evening))

Domain Space in FDL

Notation Names	Expressions in $\mathcal{FDL}$	Examples
Feature	A, C, D, D_1, D_2	Person, Female
Feature Relation	P, R, Q	hasChild
Composite Feature	$C \sqsubseteq \exists R.\top$; or $C \equiv D_1 \sqcup D_2$	Parent≡Mother⊔Father
Root Feature	$C \sqsubseteq \forall R^{-}.\bot$	Creature
Atomic Feature	$A \sqsubseteq \forall R.\bot$	White, Male
Not Feature	$C \equiv \neg D$	Mother≡¬Father
Shared Feature	$C \sqsubseteq R_1.D_1$, $C \sqsubseteq R_2.D_2$	Mother⊑Female, Mother⊑Person
Aggregative Relation	$C \equiv D_1 \sqcup D_2$	Parent≡Mother⊔Father
Instantial Relation	$C \sqsubseteq R.D$ (R is has-a relation)	Mother⊑hasChild.Person
Crossover Relation	$C \equiv D_1 \sqcap D_2$; or $D_1 \sqsubseteq R.C$, $D_2 \sqsubseteq R.C$	Mother≡Woman⊓hasChild.Person
Alternative Relation	$C \equiv D_1 \vee D_2$	Child≡ Nursery∨School∨University
OR-Relation	$C \equiv D_1 \sqcup D_2$	Parent≡Mother⊔Father
Mandatory Relation	$C \sqsubseteq R.D$	Mother⊑hasChild.Person
Optional Relation	$C \sqsubseteq (\le 1 R.\top)$	MotherofChina⊑(≤1hasChild.Person)
Num-restrict Relation	$C \sqsubseteq (\le n R.\top)$	MotherinVillage⊑(≤2hasChild.Person)
Composite Relation	$R \equiv Q_1 \sqcup Q_2$	hasChild≡isFehasChild⊔isMahasChild

Definition Space in FDL

Notation Names	Expressions in $\mathcal{FDL}$	Examples
Feature Parameter	$R.(C(a))(b)$	hasStatus.Child(yinyin)(evening)
Parameter Relation	$R(a, b)$	hasChild(zhang,yinyin)
Composite Parameter	$C(a), D(b_1), D(b_2)$ $R.(C(a))(b_1)$, $R.(C(a))(b_2)$	hasStatus.Child(yinyin)(evening) hasStatus.Child(yinyin)(4th_School)
Independent Parameter	$R.(C(a))(b)$ $\forall R^{-}.\bot(a)$	Person (zhang)
Basic Parameter	$C(a)$, $\forall R.\bot(a)$	University(tsinghua)
Shared Parameter	$C(a_1), C(a_2), D(b)$ $R.(C(a_1))(b)$, $R.(C(a_2))(b)$	hasStatus.Child(yinyin)(evening) hasStatus.Child(binbin)(evening)
Mandatory Relation	$C(a), D(b)$, $C(a) \sqsubseteq (\le 1 R.D(b))$ $R.(C(a))(b)$	Person (zhang), Child(yinyin) hasChild(zhang,yinyin)

Context Space in FDL

The Power of FDL

The Basic FDL reasoning services:

- Feature Satisfiability
- Subsumption
- Consistency
- Instance Checking

CBD problems can be solved by FDL reasoning:

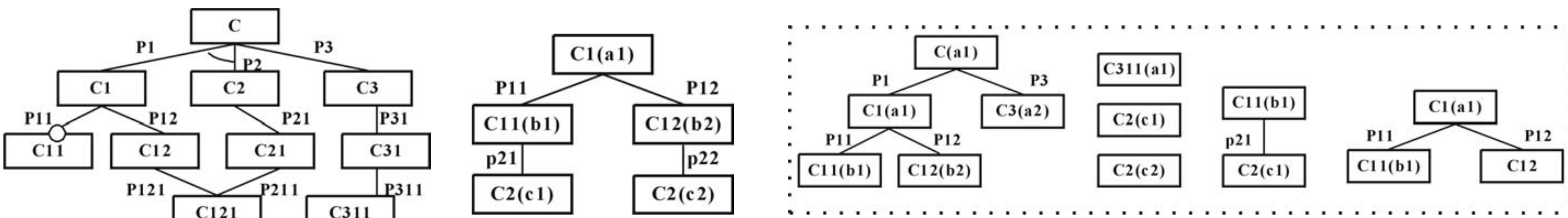
- Component categorization and retrieval
- Feature interaction
- Semantics mismatch
- Component assembly attributes prediction

The Computational Model of CBD/EA

- Representation Scheme
- Fitness Measure
- Selection Mechanism
- Genetic Operators
- Terminal Criteria

The Computational Model of CBD/EA: Representation Scheme

Coding component in FDL



(1) Domain Space (2) Problem (3) Component Population

(1) Domain Space: $\Sigma = \{ C \equiv \exists P1.C1 \sqcap \exists P3.C3 \sqcup \exists P2.C2 \sqcap P3.C3,\ C1 \equiv \exists^{\leq 1} P11.C11 \sqcap P12.C12,\ C2 \equiv P21.C21,\ C3 \equiv P31.C31,\ C12 \equiv P121.C121,\ C21 \equiv P211.C121,\ C31 \equiv P311.C311$

(2) Problem: $\Sigma = \{C1(a1), C11(b1), C12(b2), C2(c1), C2(c2), P11(a1, b1), P12(a1, b2), p21(b1, c1), p22(b2, c2)\}$

(3) Component population: $\Sigma_1 = \{ C(a1), C1(a1), C3(a2), C11(b1), C12(b2), P1(a1, a1), P2(a1, a2), P11(a1, b1), P12(a1, b2)\}$; $\Sigma_2 = \{C311(a1)\}$; $\Sigma_3 = \{C2(c1)\}$; $\Sigma_4 = \{C2(c2)\}$; $\Sigma_5 = \{C11(b1), C2(c1), p21(b1, c1)\}$; $\Sigma_6 = \{C1(a1), C11(b1), C12, P11(a1, b1), P12(a1, C12)\}$

The Computational Model of CBD/EA : Fitness Measure

(1) Knowledge-based measurement.

To computer the similarity between problem and component knowledge base;

(2)Tree-based measurement.

To computer the similarity between problem and component feature tree;

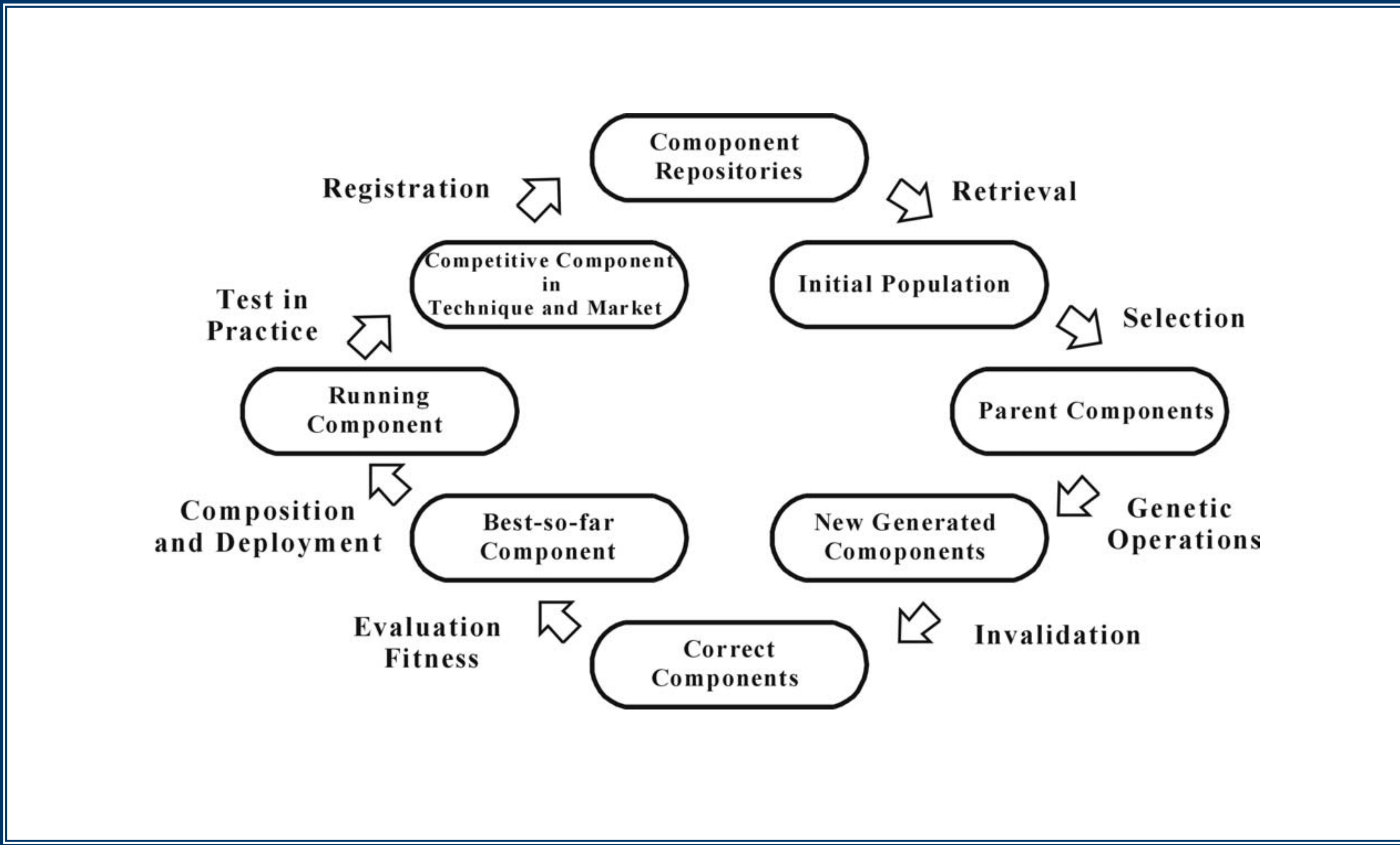
The Computational Model of CBD/EA : Selection Mechanism

In CBD/EA, a selective pressure-driven scheme is suggested according to the knowledge-based fitness, where the population is sorted into two groups in order of largest similarity (LC) value and least different (LD) value respectively.

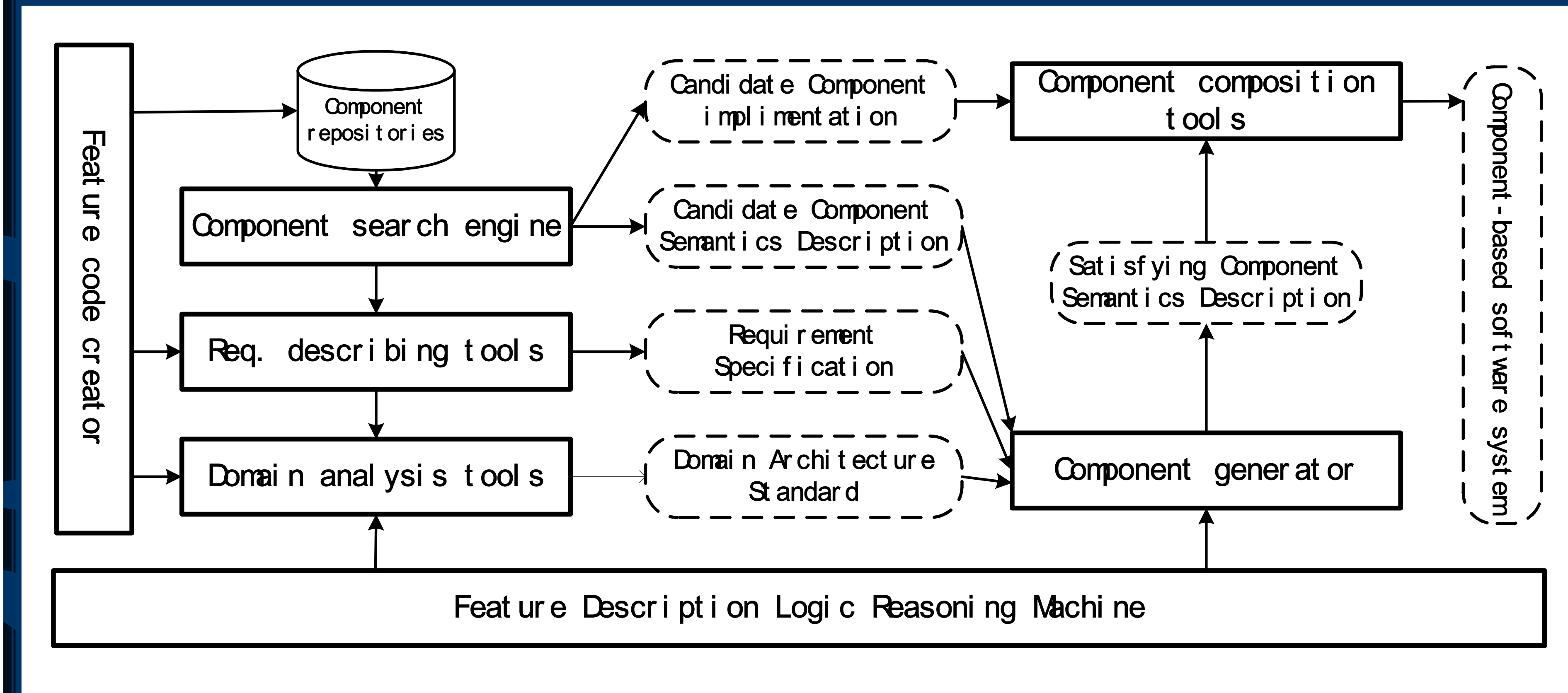
The Computational Model of CBD/EA : Genetic Operators

- Reproduction:
  - Select a component and copying an existing component code.
- Crossover:
  - Create one or more new component codes from two existing component codes.
- Mutation
  - Create a new component code from an existing component code by mutating a randomly chosen part.

The Life Cycle of Evolvable Components in CBD/EA



Future work: A Component-based Software Developing Platform



Cooperation is welcome
Thank you….

Yu Jia

Institute of Software, Chinese Academy of Science
Beijing 100080, China

jia_yu@263.net