

(crosscutting concern) [3, 8, 23].
 «
 »
 () [4]:
 (aspect-oriented software design – AOSD) [5],
 (feature-oriented software design – FOSD) [6]
 (context-oriented software development – COSD) [7],
 (separation of concerns) [4, 8]
 (lines of
 code – LOC) [9].
 () [10-12].
 [22].

2.

2.1

(AOSD, FOSD, COSD)

()

: AOSD, FOSD, COSD,

Acme ADL [16]

1. () –
 2. (layer) –
 3. (single port) –
 4. (condition port) –
 5. (binary connector interaction) –
 6. (multiple connector interaction) – 2
 7. (condition connector interaction) –
- sender → receiver,

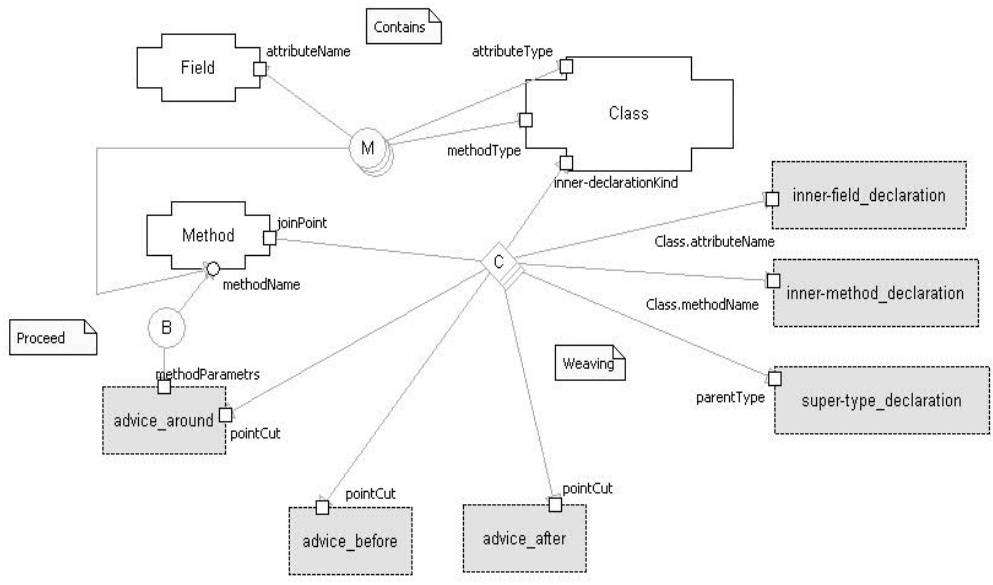
2.3

(software product line) [17, 18].

1 AOSD, 3

6

(advice_before, advice_after, advice_around),
 (inner-field_declaration, inner-method_declaration, super-
 type_declaration).



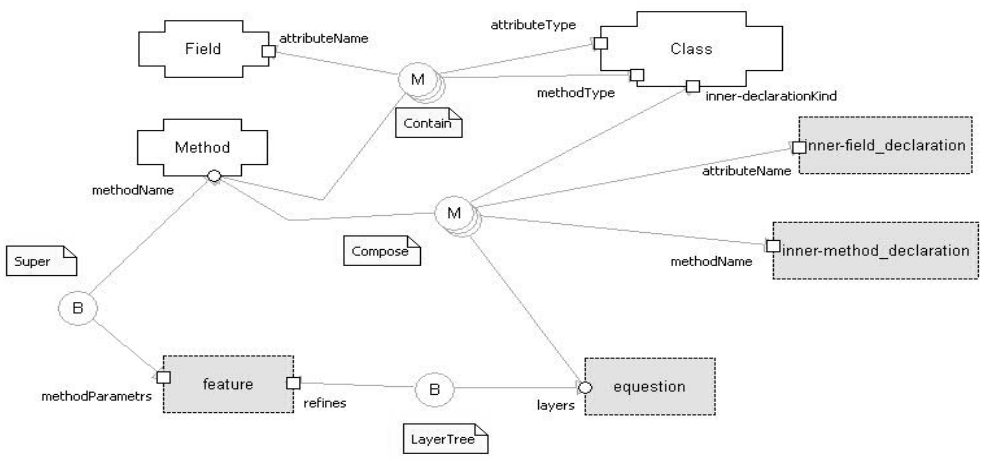
. 1.

AOSD

FOSD (. . . 2)

3

inner-method_declaration),
 (inner-field_declaration,
 (feature), equation (

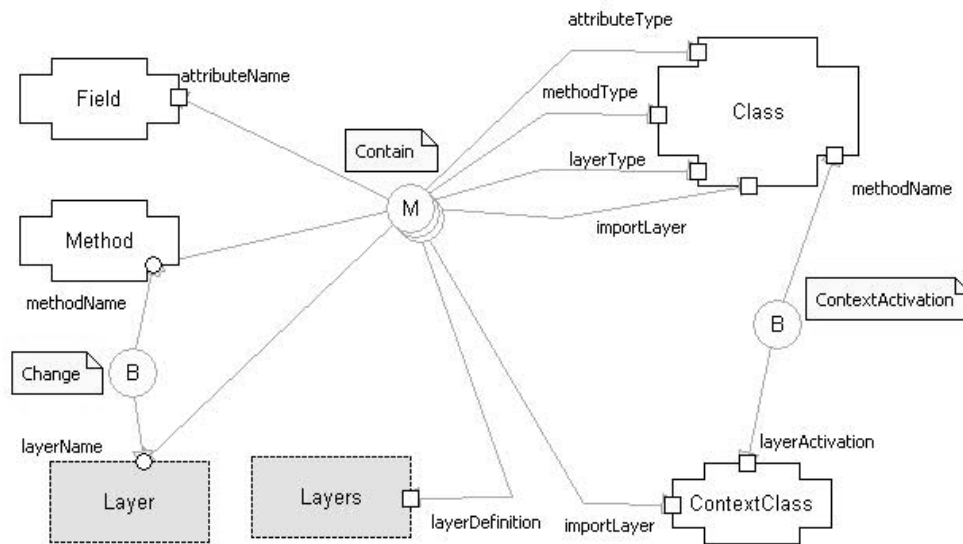


. 2.

FOSD

FOSD

OSD (. .3) 3
 : (ontextClass), (layer),
 (layers) – ,



. 3.

COSD

COSD

2.4

(Extended Architecture Primitive

Complexity - EAPC)

$$EAPC = ComponentComplexity + ConnectorComplexity + PortComplexity, \quad (2.1)$$

- ComponentComplexity – , ,
- ConnectorComplexity – , ,
- PortComplexity – , .

(2.1) , 3- ([19].

$$ComponentComplexity = \sum_{c=1}^2 w_c \#(pOOT)_c, \quad (2.2)$$

$$w_c - , \sum_{c=1}^2 w_c = 1,$$

$$\#(pOOT)_c -$$

[20].

$$SSC = \frac{\#(Cc)}{\#(Cc) + \#(Cv)}, \quad (2.3)$$

$$SSC -$$

$$\#(Cc) -$$

$$\#(Cv) -$$

$$SVC = \frac{\#(Cv)}{\#(Cc) + \#(Cv)}, \quad (2.4)$$

$$SVC -$$

$$\#(Cc) -$$

$$\#(Cv) -$$

$$(2.3),$$

$$(1-3)$$

$$SSC = \frac{9}{9+13} = 0.4,$$

$$(2.4),$$

$$SVC = \frac{13}{9+13} = 0.6.$$

$$(2.2)$$

:

$$ComponentComplexity = 0.6 \cdot \#(POOT) + 0.4 \cdot \#(OOP) \quad (2.5)$$

:

$$ConnectorComplexity = \sum_{k=1}^3 w_k \#(q)_k, \quad (2.6)$$

$$w_k - , \sum_{k=1}^3 w_k = 1,$$

$$\#(q)_k -$$

$$k-$$

:

$$PortComplexity = \sum_{r=1}^2 w_r \#(p)_r, \tag{2.7}$$

$$w_r - , \sum_{r=1}^2 w_r = 1; \#(p)_r - r- . \tag{2.6,2.7}$$

, , ,

() [21].

- (Binary Connector) $w_1=1$;
- (Multi Connector) $w_2=2$;
- (Condition Connector) $w_3=3$.

1,2)

() () [21]:

$$I = \frac{\{ \}_{max} - n}{(n-1)}, \tag{2.8}$$

$\{ \}_{max} - , \{ \}_{max} \geq n ;$
 $n - ,$

$$B = \frac{I}{BI}, \tag{2.9}$$

$BI - (, . [21]);$
 $B \leq 0,10 - .$

. 1.

	Condition Connector	Multi Connector	Binary Connector
Condition Connector	1	2	3
Multi Connector	1/2	1	3/2
Binary Connector	1/3	2/3	1

, (2.8) (2.9) : $I = 0$ $B = 0$.

:

- binary connector – $W_1=0.18$;
- multi connector – $W_2=0.27$;
- condition connector – $W_3=0.55$.

. 2.

	Condition Connector	Multi Connector	Binary Connector		
Condition Connector	1,00	2,00	3,00	6,00	0,55
Multi Connector	0,50	1,00	1,50	3,00	0,27
Binary Connector	0,33	0,66	1,00	1,99	0,18
				10,99	1
$\}_{\max} = 3$	$= 0$	$= 0,58$	$= 0$		

(2.7)

:

$$\text{ConnectorComplexity} = 0.18 \cdot \#(\text{BinaryConnector}) + 0.27 \cdot \#(\text{MultiConnector}) + 0.55 \cdot \#(\text{CaseConnector}) \quad (2.10)$$

- (SinglePort) $w1=1$;
- (ConditionPort) $w2=3$;

(.

3, 4) .

. 3.

	Condition Port	Single Port
Condition Port	1	3
Single Port	1/3	1

. 4.

	Condition Port	Single Port		
Condition Port	1,00	3,00	4,00	0,75
Single Port	0,33	1,00	1,33	0,25
			5,33	1

(2.8 – 2.9)

:

- simple port – $W_1=0.25$;
 - condition port – $W_2=0.75$.
- (2.6) :

$$PortComplexity = 0.25 \cdot \#(SinglePort) + 0.75 \cdot \#(ConditionPort), \quad (2.11)$$

$\#(SinglePort)$ – , $\#(ConditionPort)$ –
 (2.1), (2.5), (2.10) (2.11),

5.

. 5.

	Component Complexity	Connector Complexity	Port Complexity	EAPC
AOSD	4.8	1	3.75	9.55
FOSD	3.6	0.9	3.5	8.0
COSD	3.0	0.63	3.75	7.38

3.

- : 1) (AOSD, FOSD, COSD),
- ; 2)
- ; 3)
- ()
- AcmeADL; 4)
- ; 5)

1.
//
" " – : " " . – 2010. – 67. – .116-125.
2. Robert E. Filman et al. Aspect-Oriented Software Development. – Addison Wesley Professional, 2004. – 800 p.
3.
-
//
(Problems in Programmimg. Scientific Journal). ISSN 1727 – 4907 – .:
.- 2010. – 2-3 (. . . .). – . 252-260.
4. Adam Przybyłek. Post object-oriented paradigms in software development: a comparative analysis // Proceedings of the International Multiconference on Computer Science and Information Technology, 2007. – pp. 1009-1020.
5. Aspect-Oriented Software Development [. . . .]. – <http://aosd.net>.
6. Feature-Oriented Software Development [. . . .]. – <http://fosd.de>.
7. Context-Oriented Software Development [. . . .]. – <http://www.hpi.uni-potsdam.de/hirschfeld/cop/events>.
8. Siobhán Clarket et al. Separating Concerns throughout the Development Lifecycle // Intl Workshop on AspectOriented Programming ECOOP, 1999.
9. Sven Apel. How AspectJ is Used: An Analysis of Eleven AspectJ Programs // Journal of Object Technology. – 2010. vol. 9, no. 1. – pp 117–142.
10.
//
– 2010. – 2.
11.
Web- –
81. – 2010.
12.
//
.- 2009. – 3. – . 40-47.
13. David Garlan, Robert Monroe, David Wile ACME: An Architecture Description Interchange Language // CASCON '97 Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research. – 1997.
14. Ackbar Joolia, Thais Batista et. al. Mapping ADL Specifications to an Efficient and Reconfigurable Runtime Component Platform // WICSA '05 Proceedings of the 5th Working IEEE/IFIP Conference on Software Architecture – 2005. – p.28.
15. Miguel Goulão and Fernando Brito e Abreu. Bridging the gap between Acme and UML 2.0 for CBD". In Proc. of Specification and Verification of Component-

