

# **REVOLUTION OR EVOLUTION? A COMPARISON OF OBJECT-ORIENTED AND STRUCTURED SYSTEMS DEVELOPMENT METHODS**

Oleh Sumit Sircar, Sridhar P. Nerur, Radhakanta Mahapatra  
MISQ Vol.25 No 4, Desember 2001

---

Kelompok : 143  
Nama : Ardhita Maharindra  
NPM : 120200015X

## **Kata Kunci**

---

*IS Development, structured development approach, object-oriented approach, software development methodologies, author co-citation analysis.*

## **Ringkasan**

---

### **Masalah**

Dalam beberapa tahun terakhir (saat artikel ini ditulis) perhatian komunitas pengembang *software* tertuju pada keuntungan-keuntungan yang ditawarkan oleh teknologi *Object-Oriented* (OO). OO dikatakan dapat mengatasi beberapa masalah yang terdapat pada pengembangan *software* sebelumnya (*structured approach*) seperti peningkatan produktivitas, lebih sedikitnya masalah *maintenance*, pengembangan sistem yang lebih cepat, dan peningkatan kualitas dari sistem.

### **Tujuan**

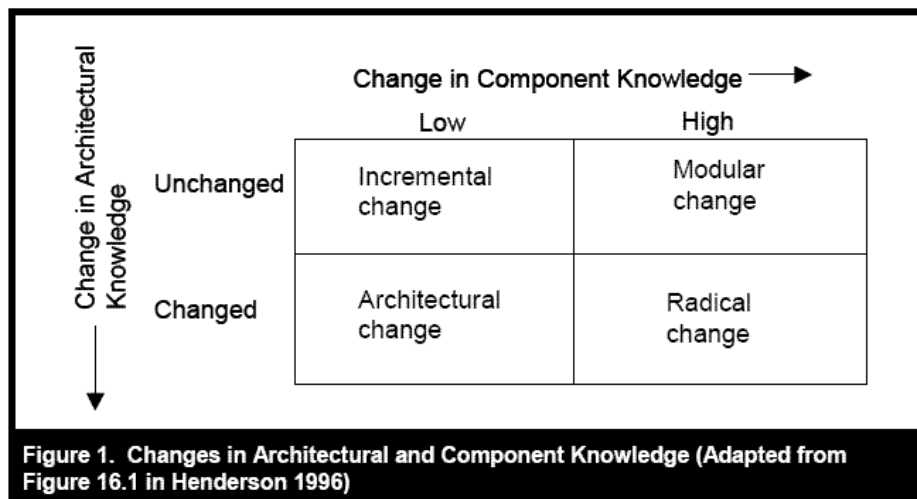
Terdapat tiga tujuan dalam riset ini yaitu :

1. Menguraikan konsep hubungan antara OO dan *structured methods* menggunakan beberapa *framework* riset.
2. Menyatukan beberapa pandangan mengenai sifat dari perubahan menuju pendekatan OO.
3. Menawarkan panduan kepada organisasi yang mengalami transisi ke OO *computing*.

### **Asumsi dan Hipotesis**

Dalam riset ini penulis hendak mengetahui sifat perubahan dari *structured method* ke OO *method*. Oleh karena itu penulis menggunakan *framework* kategori perubahan yang dicetuskan oleh Henderson dan Clark pada tahun 1990. *Framework* ini memiliki dua dimensi dalam perubahan yaitu perubahan komponen dan perubahan arsitektur. Komponen adalah bagian-bagian dari sistem yang menyusun keseluruhan sistem. Arsitektur adalah spesifikasi tentang bagaimana komponen-komponen saling berinteraksi membentuk suatu sistem.

Terdapat empat jenis perubahan dalam framework ini. Yang pertama adalah *incremental change* dimana terdapat sedikit perubahan komponen ilmu dan tidak terdapat perubahan arsitektur. Perubahan ini disebut juga perubahan yang *evolutionary*. Perubahan ke dua adalah *modular change* dimana terdapat banyak perubahan komponen namun tidak terdapat perubahan arsitektur. Jenis perubahan yang ke tiga adalah *architectural change* dimana terdapat perubahan arsitektur namun hanya sedikit terjadi perubahan komponen. Jenis perubahan yang terakhir adalah *radical change* dimana terdapat banyak perubahan arsitektur dan komponen. Perubahan ini disebut juga perubahan *revolutionary*. Untuk lebih jelasnya tentang framework ini dapat dilihat pada gambar 1.



Gambar 1 Framework perubahan oleh Henderson dan Clark

Untuk mengetahui jenis perubahan dari *structured method* ke *OO method* maka terlebih dahulu dibahas mengenai proses-proses yang ada pada kedua metoda pengembangan tersebut. Pada tahap analisis dari *structured method* dilakukan pendefinisian proses yang dilakukan dengan *Data Flow Diagram* dan pendefinisian data yang dilakukan dengan merancang *Entity Relationship Diagram*. Kemudian pada tahap disain dari *structured method* dilakukan pembuatan detail spesifikasi program dan skema *database*. Pada tahap implementasi dilakukan implementasi program, implementasi *database*, integrasi dan *testing*. Sedangkan untuk *OO method* pada tahap analisis dibentuk *usecases* dan *object models (class diagram)*. Pada tahap disain dilakukan pembuatan *design class specification* dan *sequence diagram*. Pada tahap implementasi *OO method* dilakukan implementasi *class*, implementasi *database* untuk *object yang persistent*, integrasi dan *testing*.

Dari proses-proses pada kedua metoda pengembangan software yang telah dijabarkan sebelumnya maka dapat dinyatakan dua proposisi yaitu :

**Proposisi 1** : Transisi dari tahap analisis dan disain pada *structured systems* ke tahap analisis dan disain pada *object-oriented* merupakan perubahan *architectural change*.

**Proposisi 2** : Transisi dari tahap implementasi pada *structured systems* ke tahap implementasi pada *object-oriented* merupakan perubahan *incremental change*.

### Metoda Penelitian

Metoda penelitian yang digunakan adalah *co-citation analysis*. Metoda ini digunakan untuk mengamati struktur intelektual dari bidang *ilmu* dan untuk mengidentifikasi kedekatan antara sub-sub bidang tersebut. Metoda ini dimulai dengan memberikan *list* dari pengarang-pengarang yang telah berkontribusi dalam suatu disiplin ilmu. Kemudian dilakukan perhitungan *co-citation* untuk tiap pasangan pengarang (misalnya, jumlah paper yang mengutip hasil kerja dari kedua pengarang). Data tersebut kemudian diolah lebih lanjut dengan berbagai metoda statistik untuk keperluan analisis lebih lanjut.

### Hasil Penelitian

Pada penelitian ini dilakukan *co-citation analysis* terhadap 30 pengarang yang memiliki kontribusi tinggi pada bidang *software development*. Dari hasil analisis tersebut didapatkan data mengenai jumlah artikel yang mengutip pasangan pengarang pada tabel 1.

	BOEHM	BOOCH	BROOKS	BUHR	COAD
BOEHM	216.5	61	169	7	23
BOOCH	61	181	31	22	105
BROOKS	169	31	155.5	4	13
BUHR	7	22	4	23.5	3
COAD	23	105	13	3	130.5

Figure 2. Sample Raw Co-citation Matrix

Table 1 Matriks Raw Co-Citation

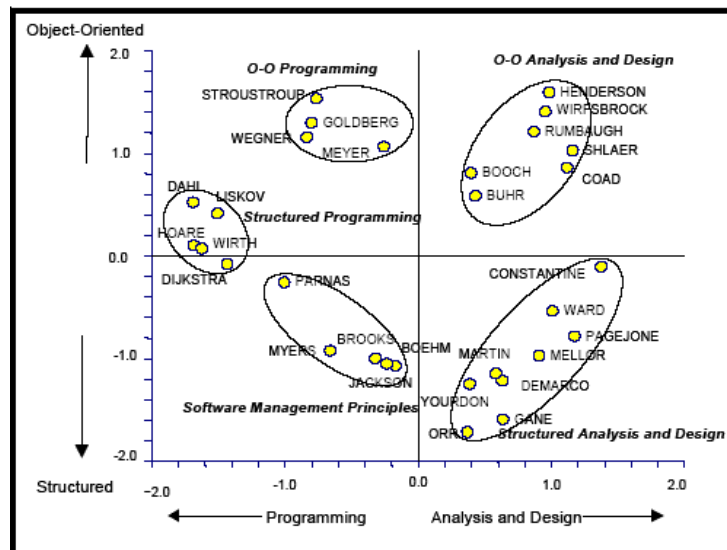
Setelah itu dilakukan *factor analysis* untuk mengetahui pengelompokan dari pengarang-pengarang yang disubsetkan dalam 5 faktor (kategori bidang). Hasil dari analisis ini ditampilkan pada tabel 2.

Factor Name	Structured Analysis and Design		Object-Oriented Analysis and Design		Structured Programming		Object-Oriented Programming		Software Management Principles	
Eigenvalue	7.61301		6.69034		5.60226		2.02078		1.70030	
% Total Variance	25.4		22.3		18.7		6.7		5.7	
Cumulative % Variance	25.4		47.7		66.4		73.1		78.8	
Loading > 0.40	Gane	0.85	Coad	0.93	Hoare	0.93	Goldberg	0.95	Brooks	0.92
	DeMarco	0.84	Rumbaugh	0.91	Dijkstra	0.93	Stroustrup	0.93	Boehm	0.91
	Ward	0.80	Shlaer	0.87	Wirth	0.87	Meyer	0.88	Myers	0.72
	Yourdon	0.79	Wirfs-Brock	0.85	Liskov	0.79	Wegner	0.84	Martin	0.56
	Page-Jones	0.79	Booch	0.81	Parnas	0.76	Dahl	0.61	Yourdon	0.56
	Mellor	0.78	Henderson-Sellers	0.73	Dahl	0.60	Booch	0.50	Parnas	0.52
	Orr	0.62	Buhr	0.50			Liskov	0.48	DeMarco	0.45
	Martin	0.59							Jackson	0.42
	Jackson	0.51								

Table 2 Hasil Factor Analysis

Kemudian dilakukan *cluster analysis* dengan metoda Ward's yang dilakukan untuk mengidentifikasi *cluster* dari pengarang-pengarang yang sejenis. Hasil dari analisis ini adalah 5 *cluster* yang konsisten dengan faktor-faktor yang diidentifikasi pada *factor analysis*. Dari analisis ini ditemukan bahwa *structured programming* secara konsep lebih dekat ke *OO programming* dibandingkan kepada tahap analisis dan disain dari *structured method*.

Analisis selanjutnya yang dilakukan adalah *multidimensional scalling* (MDS) untuk menggambar grafik dari penyebaran pengarang dalam ruang intelektual, MDS dilakukan dengan prosedur ALSCAL di SPSS. *Cluster-cluster* yang menjadi pembatas pada grafik MDS dibuat berdasarkan lima *cluster* yang didefinisikan pada *cluster analysis*. Pemetaan MDS digambarkan pada gambar 2.



Gambar 2 Pemetaan MDS

### Kesimpulan

Dari hasil analisis-analisis yang telah dilakukan didapatkan bahwa OO dan *structured systems development* berbeda secara dalam beberapa hal namun masih menggunakan konsep yang umum. Perbedaan dari OO dan *structured method* lebih terletak pada tahap analisis dan disain dibandingkan dengan tahap implementasi. Pada tahap analisis dan disain terdapat perbedaan tentang bagaimana proses dan data didefinisikan. Dimana pada *structured method* proses dan data dimodelkan secara independent sedangkan pada OO *method* proses dan data dimodelkan secara terkait. Sedangkan komponen sistem informasi yang digunakan tidak berubah dalam kedua metoda ini yaitu logika proses dan data. Oleh karena itu transisi dari tahap analisis dan disain dari structured ke OO termasuk *architectural change*.

Pada tahap implementasi tidak terdapat perubahan secara arsitektur namun terdapat sedikit perubahan komponen seperti bahasa pemrograman, misalnya dari C ke C++. Oleh karena itu transisi pada tahap implementasi dari *structured method* ke *OO method* merupakan perubahan *incremental change*. Maka secara keseluruhan transisi dari *structured* ke *OO* tidak bisa dibilang *evolutionary* maupun *revolutionary* melainkan merupakan perubahan *architectural change*.

Temuan ini memiliki implikasi bagi organisasi terutama IS *manager* yang sedang dalam proses migrasi ke pendekatan *OO*. Bantuan terhadap tim *developer* harus lebih ditingkatkan pada pengadopsian tahap analisis dan disain dari *OO*. Perubahan dari arsitektural *knowledge* (definisi menurut Henderson) dapat difasilitasi dengan kombinasi dari *training*, *mentoring* dan pengadopsian *CASE tools* yang tepat.

### **Komentar**

Menurut saya artikel ini sangat menarik, bukan hanya karena pembahasan dan tema yang diangkat namun karena metoda penelitian yang menurut saya sangat rumit dan unik. Jenis metoda *co-citation analysis* baru pertama kalinya saya temukan dalam artikel jadi saya cukup tertarik. Urutan pendefinisian masalah, asumsi-asumsi yang digunakan serta metoda dijelaskan dengan cukup jelas. Namun pada bagian analisis dari hasil penelitian dengan metoda *factor analysis*, *cluster analysis* dan *MDS* belum saya pahami sepenuhnya.

### **Referensi**

---

Bayer, E. A., Smart, C. J., and McLaughlin, W. G..Mapping Intellectual Structure of a Scientific Subfield through Author Co-citations,. *Journal of the American Society for Information Science* (41:6), 1990, pp. 444-452.

Henderson, R. M. .Technological Change and theManagement of Architectural Knowledge,. In *Organizational Learning*, M. D. Cohen and L. S. Sproull (eds.), Sage Publications, Thousand Oaks, CA, 1996, pp. 359-375.

Jacobson, I., Ericsson, M., and Jacobson, A. *The Object Advantage.Business Process Reengineering with Object Technology*, Addison-Wesley, Reading, MA, 1995.

McCain, K. W. .The Author Cocitation Structure of Macroeconomics, *Scientometrics* (5), 1983, pp. 277-289.

White, D. H. .Author Co-Citation Analysis: Overview and Defense,. *Scholarly Communicationand Bibliometrics*, C. L. Borgman (ed.), Sage Publications, Thousand Oaks, CA, 1990, pp. 84-106.