

MODUL 8 :

Object Interaction

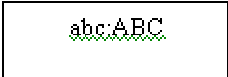
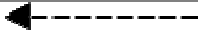
Contents

8.1	Pendahuluan.....	2
8.2	Elemen pada Sequence Diagram	3
8.2.1	Partisipant.....	3
	Interaksi Object dan Collaborasi	3
8.3	Interaksi Sequence Diagram.....	4
8.3.1	Konsep dan Notasi.....	5
	Boundary dan control Class.....	7
	Object Destruction.....	7
	Reflexive Message.....	8
	Focus of Control.....	9
	Replay Message	10
	Object Selector	10
	Interaction Operation	12
8.3.2	Menangani Kompleksitas.....	13
	Interaction fragment.....	14
	Lifelines untuk subsystems atau kumpulan Object	16
	Continuations	19
8.4	Petunjuk pembuatan Sequence Diagram.....	19
	Referensi.....	21

8.1 Pendahuluan

Sequence diagram merupakan sebuah diagram yang menggambarkan interaksi antar objek di dalam sebuah system. Interaksi tersebut berupa message yang digambarkan terhadap waktu. Sequence diagram terdiri dari dimensi horizontal (objek-objek) dan dimensi vertical (waktu). Diagram ini juga menggambarkan urutan even yang terjadi. Dan lebih detail dalam menggambarkan aliran data, termasuk data atau behavior yang dikirimkan atau diterima. Namun, diagram ini kurang mampu menjelaskan detail dari sebuah algoritma, seperti loop, branching.

Tabel 8.1. Komponen pada Sequence Diagram

	Actor	Menggambarkan sebuah orang atau entitas eksternal yang berinteraksi dengan sistem
	Object	Menggambarkan sebuah objek dalam sebuah system atau salah satu komponennya. Objek merupakan sebuah instansi dari sebuah class. Bagian “abc” merupakan nama objek dan “ABC” merupakan nama class
	Lifeline	Menggambarkan daur hidup sebuah objek
	Activation bar	Menggambarkan durasi/lamanya pengerjaan sebuah pesan (message).
	Message	Sebuah message sederhana antara elemen header (bias berupa synchronous atau asynchronous). Dan juga mengindikasikan komunikasi antara objek.
	Synchronous Message	Menggambarkan message ke elemen (objek) lain, dimana message ini mengaktifkan sebuah proses dan sampai selesai, baru bisa mengirimkan sebuah message baru.
	Return	Suatu hasil kembalian sebuah operasi (feedback dari sebuah message)

	Asynchronous Message	Menggambarkan message ke elemen (objek) lain, dimana message ini mengaktifkan sebuah proses dan ditengah-tengah proses, pengirim message bisa mengirimkan message baru.
	Destroy Element	Menggambarkan penghancuran sebuah elemen header (objek) setelah selesai sebuah operasi
	Destroy Message	Menggambarkan penghancuran sebuah elemen header (objek) sebagai akibat dari sebuah message dari elemen/objek lain

8.2 Elemen pada Sequence Diagram

8.2.1 Partisipant

Interaksi Object dan Collaborasi

Pada saat sebuah objek berkomunikasi dengan mengirimkan pesan ke objek yang lain maka sebuah operasi yang dimiliki oleh objek penerima dipanggil oleh objek pengirim (*send message*). Misalkan terjadi komunikasi antara class Campaign dan Class Advert. Class Campaign ingin mengetahui berapa cost untuk masing-masing Advert. Dalam bahasa pemrograman ditulis sebagai berikut.

```
currentadvertCost = anAdvert.getCost ()
```

sebuah objek dari class advert di indentifikasi dengan nama anAdvert dan respon diidentifikasi sebagai return value yang disimpan dalam variabel currentadvertCost. Satau hal yang mungkin terjadi adalah sulit untuk menentukan apa pesan (message) yang di kirim oleh masing-masing object. Pada kasus diatas jelas bahwa operasi getCost terdapat pada class Advert.

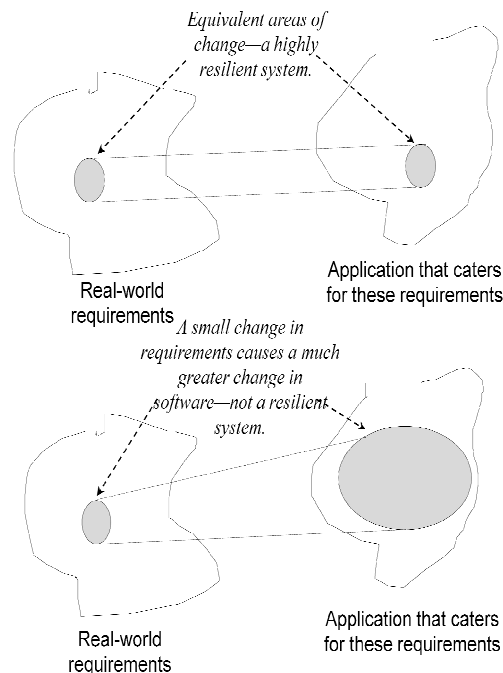


Gambar 8.1. Object Messagin

Sumber:

Simon Bennet, Steve McRobb and Ray Farmer, *Object Oriented Systems Analysis and Design Using UML*, Edisi 3. ; McGraw Hill, 2006.

Pada analisa berorientasi objek, sebaiknya fungsionalitas dari setiap kelas tersebar merata. Hal tersebut bukan berarti setiap kelas memiliki tanggung jawab yang sama melainkan tergantung dari level class. Pada saat fungsionalitas tersebar merata maka akan memudahkan untuk mengembangkan, melakukan tes serta memelihara. Class yang relatif kecil sangat potensial untuk digunakan kembali dibandingkan dengan class yang besar dan kompleks.



Gambar 8.2 Ressilience of a Design

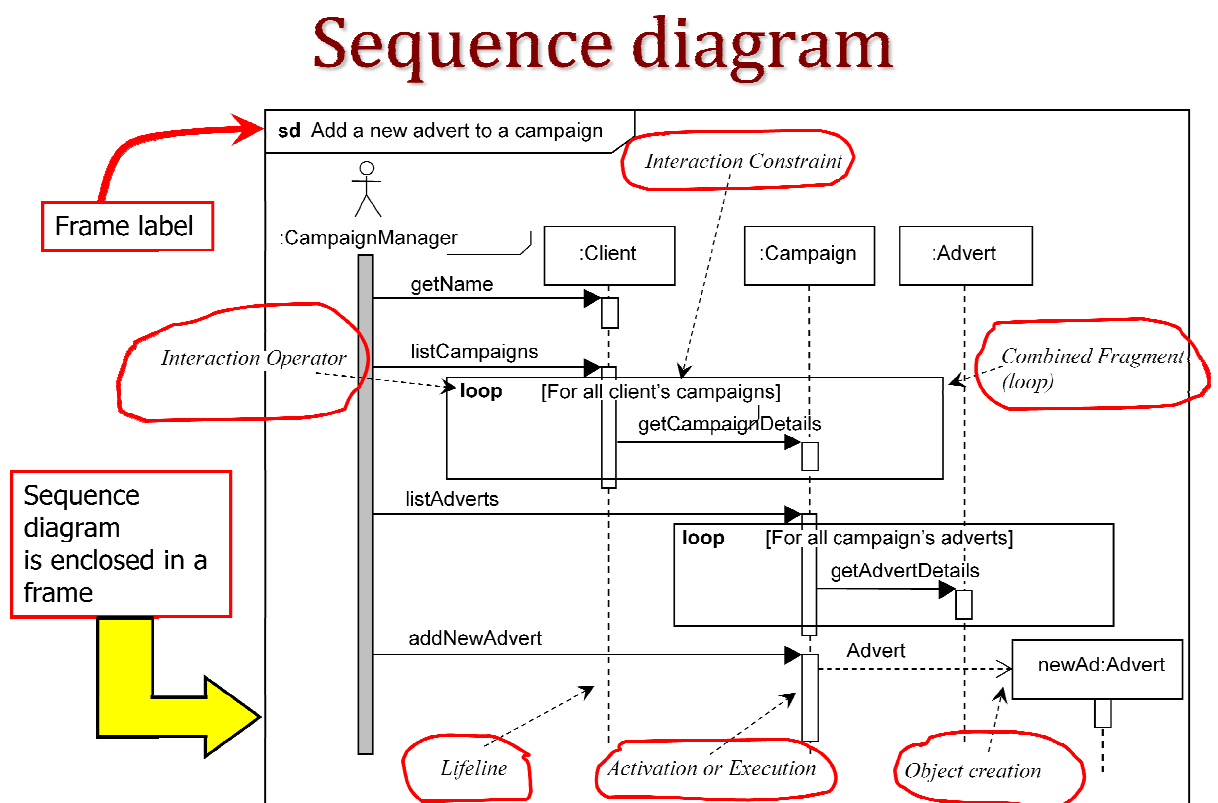
Fungsionalitas yang merata pada class akan berdampak pada sistem dimana penanganannya akan lebih mudah jika ada perubahan requirement sehingga perubahan aplikasi tidak terlalu. Besar.

8.3 Interaksi Sequence Diagram

Merupakan salah satu diagram interaksi yang terdapat pada UML. Secara sematik hampir mirip dengan communication diagram jika kasusnya sederhana. Sequence Diagram menggambarkan komunikasi antar objek sesuai dengan urutan waktu. Sequence diagram bisa digambarkan dengan level yang lebih rinci sesuai dengan tujuan yang akan dicapai.

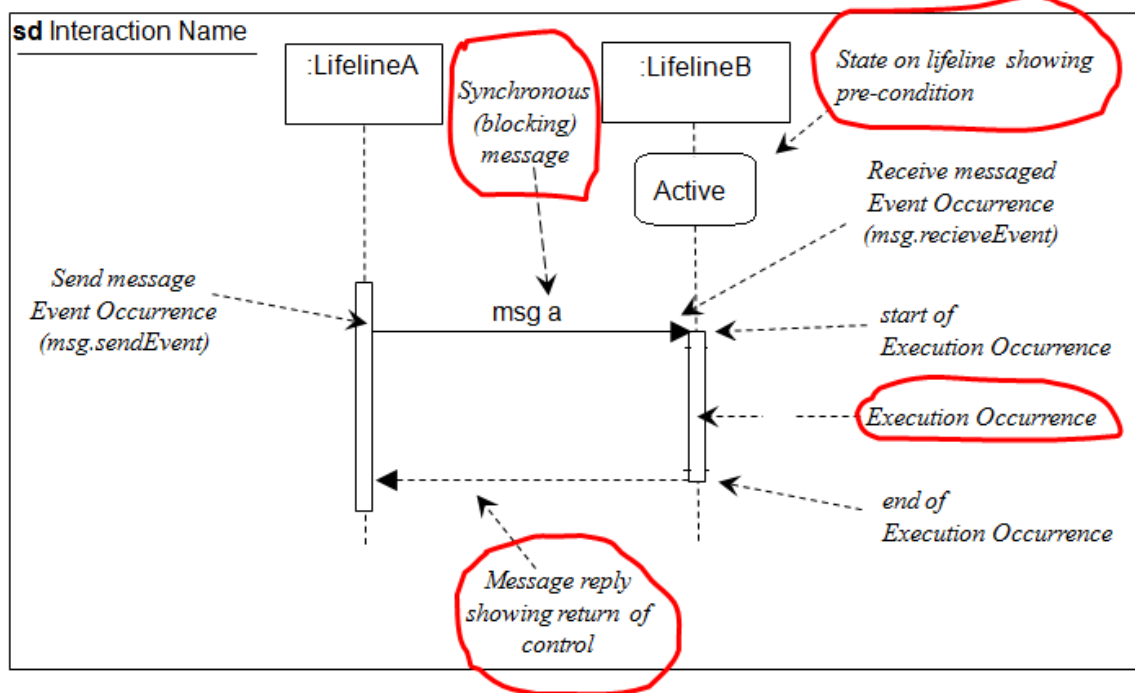
8.3.1 Konsep dan Notasi

Setiap objek pada diagram sequence direpresentasikan dengan *lifeline* berupa garis vertikal sebagai simbol objek yang ditempatkan pada bagian atas diagram. Sebuah Message digambarkan dengan garis horisontal berpanah dari satu lifeline ke yang lainnya yang diberi nama sesuai dengan nama messagenya. Setiap message dapat diberi nomor untuk menunjukkan urutan dari pengiriman message namun hal tersebut tidak terlalu penting jika message tersebut ditempatkan sesuai dengan posisinya sesuai dengan urutan waktu. Pada saat message dikirim ke sebuah objek, maka objek tersebut memanggil sebuah operasi yang dimiliki oleh objek yang dikirim message.



Gambar 8.3 Sequence Diagram untuk Use case Add a new advert to a campaign

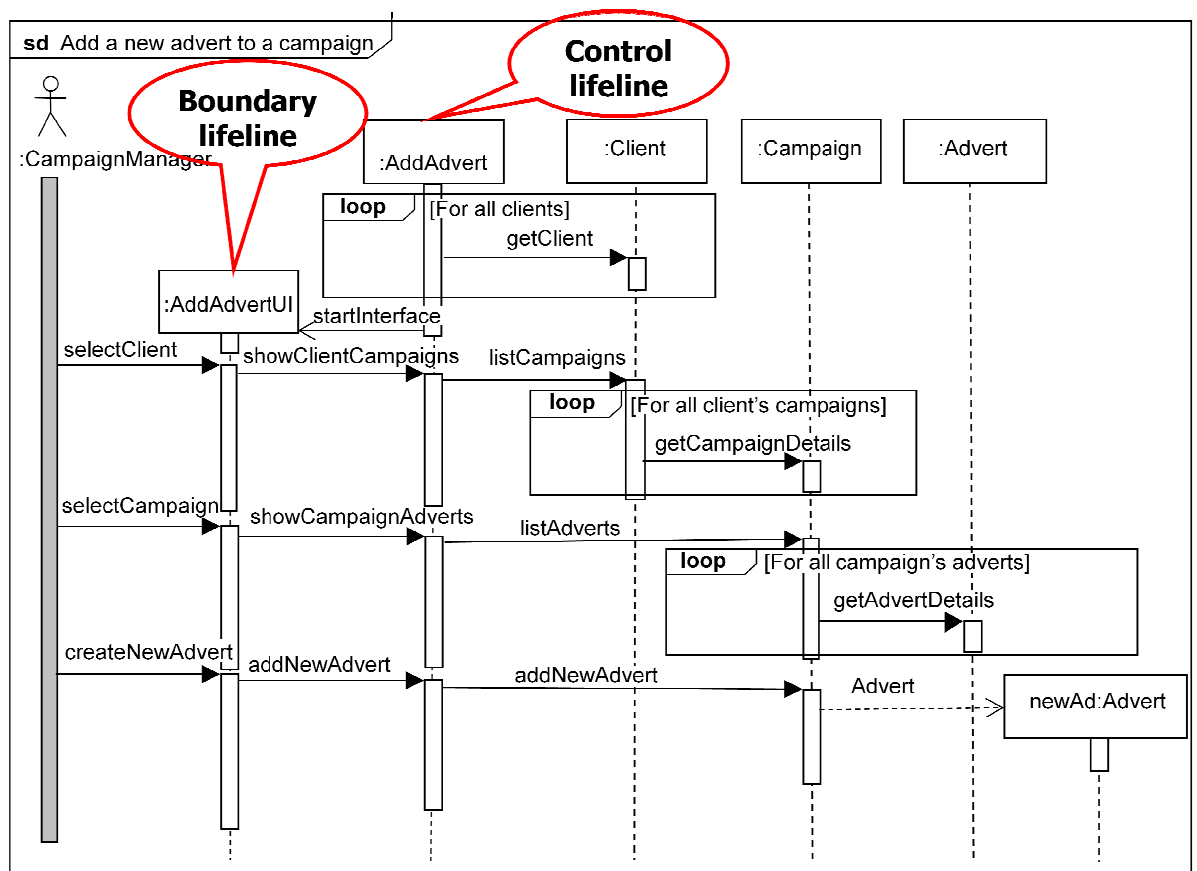
Sebuah pengulangan ditandai dengan kotak persegi panjang yang diberi label loop pada bagian atas. Bagian loop tersebut akan dijalankan jika kondisi dari constraint yang membatasi bernilai benar. Sebuah message Synchronous digambarkan dengan garis panah penuh.



Gambar 8.4 Sequence Diagram untuk menunjukkan pesan dan eksekusi event dan state

Boundary dan control Class

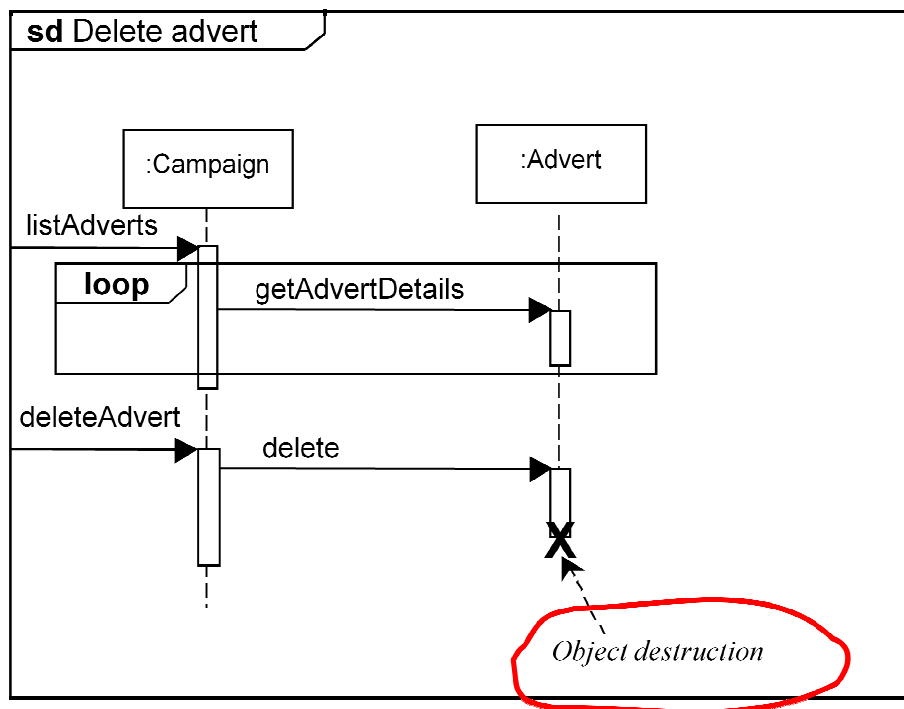
Beberapa UseCase memiliki setidaknya 1 buah object **boundary** yang mengatur interaksi antara Actor dan system. Pada *sequence diagram* di representasikan dengan lifeline **AddAdvertUI**. **Control object** direpresentasikan dengan lifeline **AddAdvert** yang mengatur semua komunikasi object tersebut.



Gambar 8.5 Sequence Diagram untuk Use case Add a new advert to a campaign dengan *boundary dan control classes*

Object Destruction

Sebuah object bisa saja dibentuk atau dihilangkan pada saat terjadi interaksi. Pada sequence diagram untuk menghilangkan object dilakukan dengan menggunakan tanda silang (X).

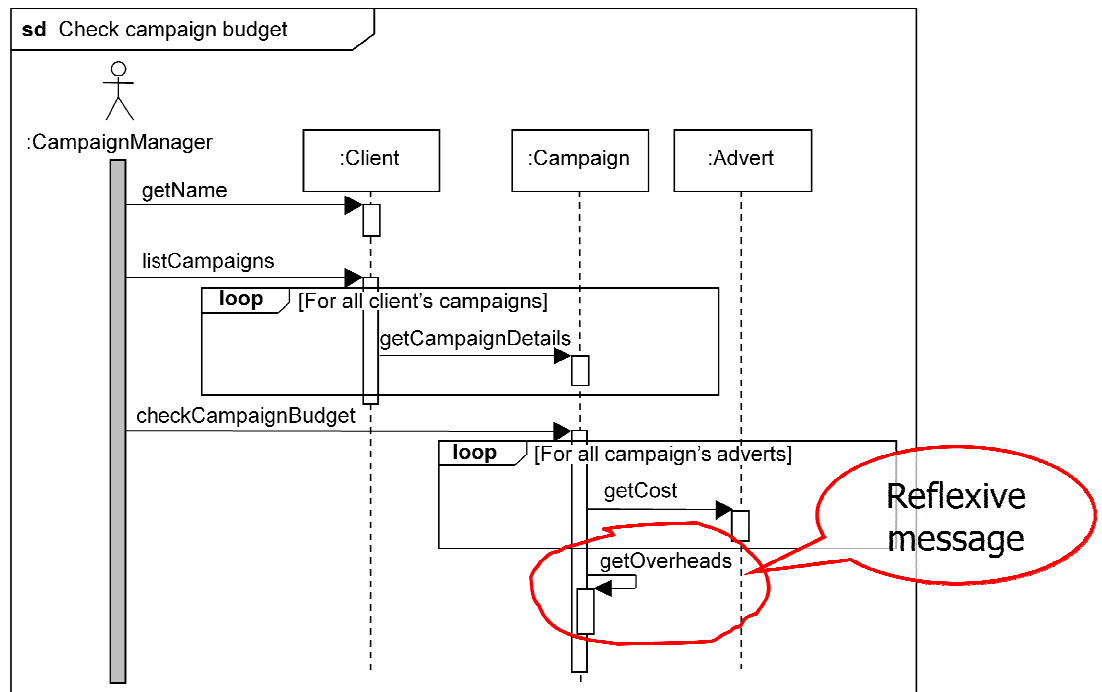


Gambar 8.6 Obejct destructions

Refrexive Message

Suatu object bisa memanggil untuk meminta operasi pada dirinya sendiri. Pada kasus dibawah ini ketika object Campaignmanager meminta berapa biaya yang dibutuhkan pada setiap **campaign** (*checkCampaignBudget*) maka object **Campaign** meminta pengeluaran dari setiap **Adverd** (*getCost*) dengan mencari berapa pengeluaran setiap Adverd pada *Campaign* yang dimaksud. Hal ini dilakukan secara iterasi untuk setiap campaign (*loop*). Selain dari cost setiap adverd, budged dari campaign diperoleh dari overhead dimana overhead adaah operasi yang dimiliki oleh campaign itu sendiri. Proses pemanggilan operasi ke diri sendiri disebut **reflexive message**.

Reflexive Messages

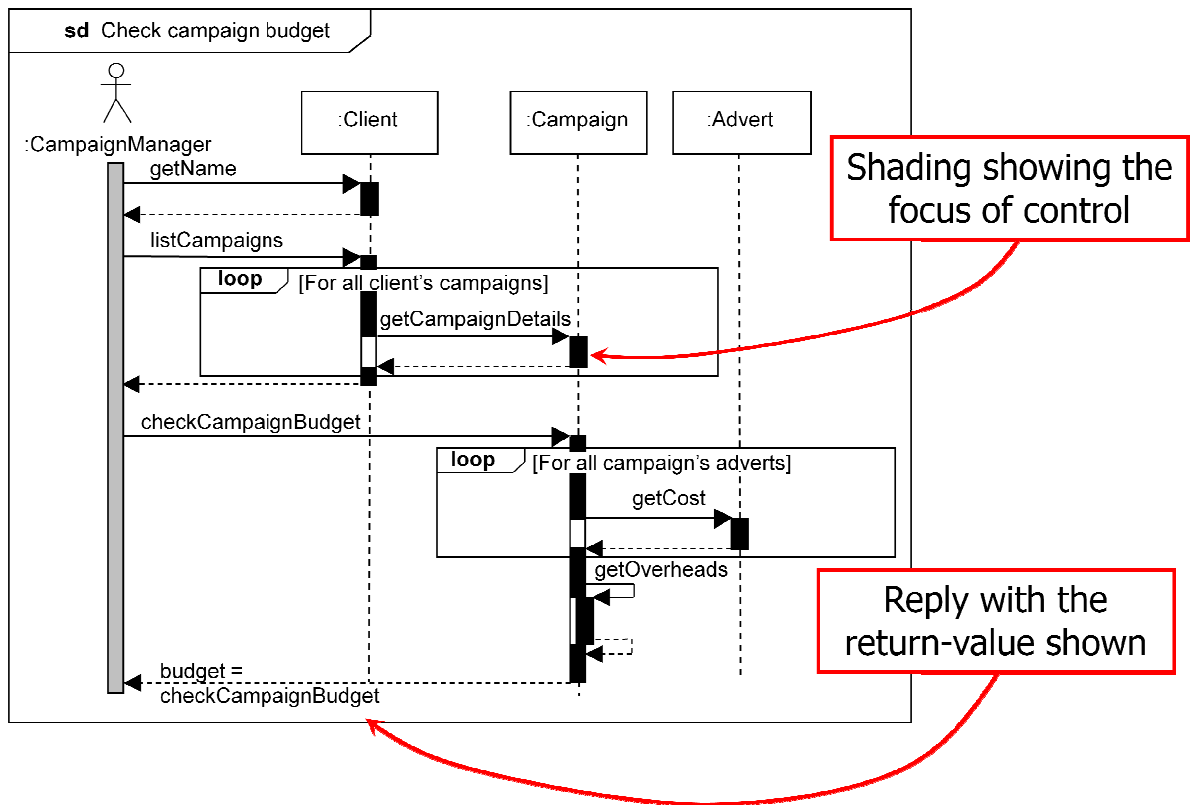


Gambar 8.7 Sequence Diagram dengan Refelexive Message

Focus of Control

Menunjukkan waktu selama proses yang terjadi pada object. Bagian dari kegiatan yang terdapat pada focus of control ditunjukkan dengan garis persegi panjang.

Focus of Control



Gambar 8.8 Sequence Diagram dengan Refelexive Message

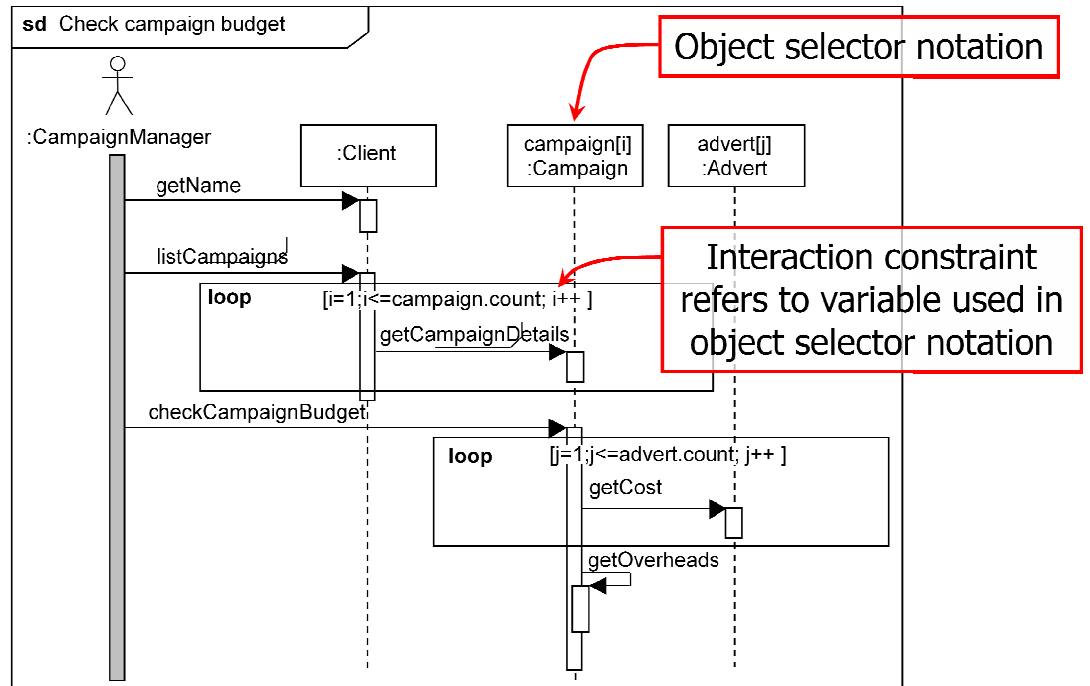
Replay Message

Sebuah pesan balasan (replay message) ditampilkan dengan menggunakan tanda panah putus-putus. Namun hal tersebut sifatnya optional.

Object Selector

Sequence diagram dapat digunakan untuk menggambarkan obyek yang dipilih pada iterasi tertentu (dari sebuah loop yang beinterasi dari 1 ke nilai campaign.count).

Object Selector Notation

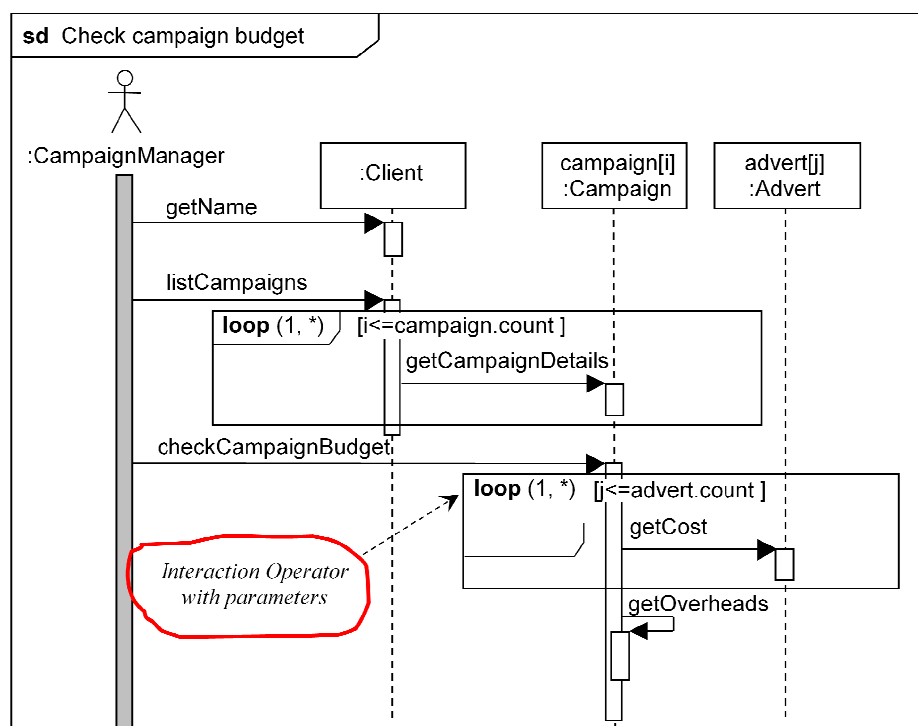


Gambar 8.9 Sequence Diagram dengan Object Selector Notation

Interaction Operation

Sequence diagram juga dapat menggambarkan operator interaksi loop yang digambarkan dengan parameter. Parameter pertama adalah jumlah minimum iterasi dan parameter kedua adalah jumlah maksimum iterasi.

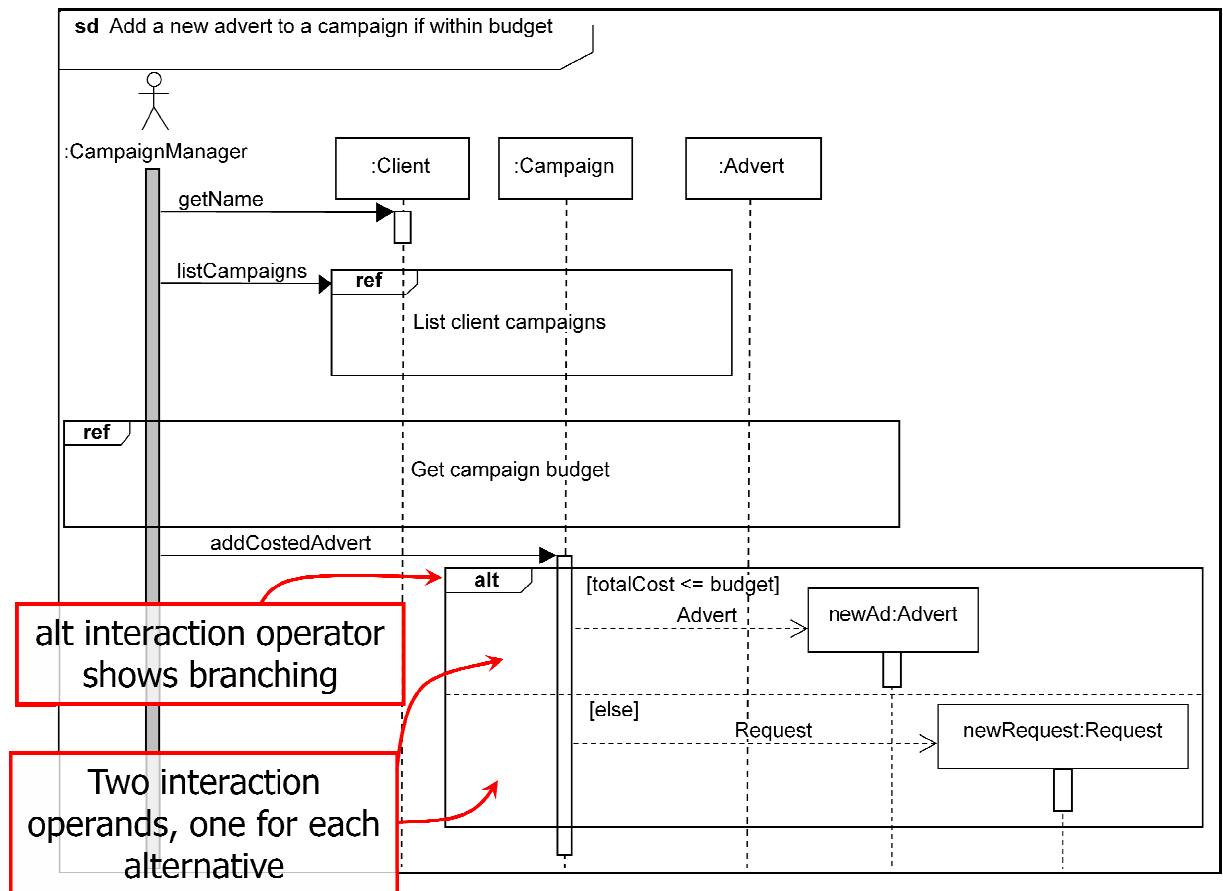
Interaction Operators



Gambar 8.10 Sequence Diagram dengan interaction operator with parameter

Branching

Beberapa interaksi pada sequence diagram dapat memiliki dua atau lebih *alternative execution pathways*. Setiap alternatif menggambarkan suatu cabang pada urutan yang mungkin dari event untuk use case yang diwakilinya.



Gambar 8.11 Sequence Diagram yang menunjukkan percabangan (branching)

8.3.2 Menangani Kompleksitas

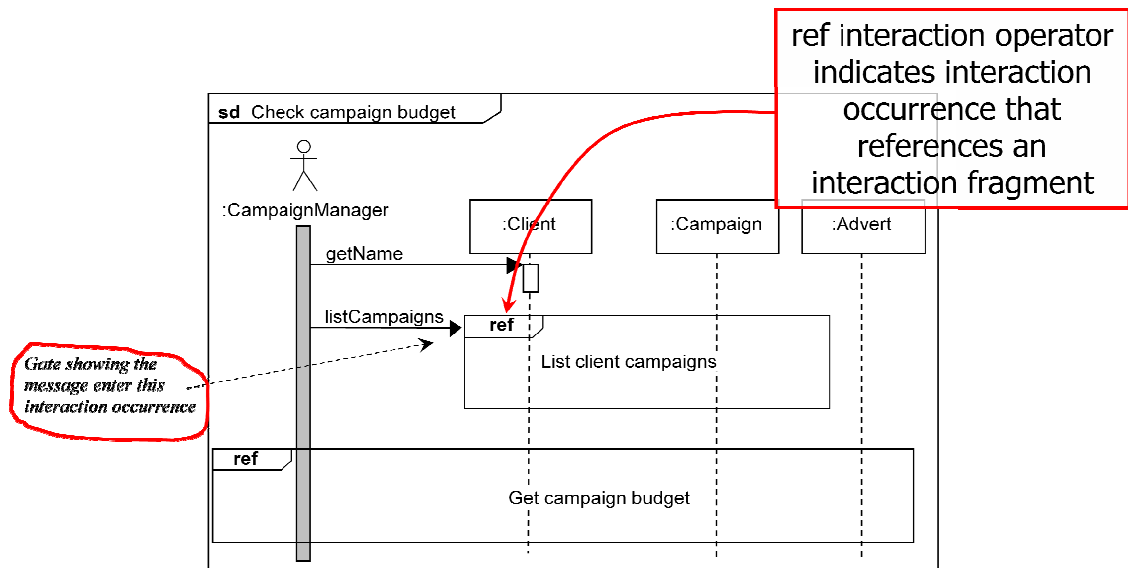
Untuk mengendalikan kompleksitas sequence diagram dapat dilakukan dengan beberapa cara antara lain :

- Interaction fragments
- Lifelines untuk subsystems atau kumpulan Object
- Continuations
- Interaction Overview Diagrams

Interaction fragment

Disini interaksi dimodelkan dengan lebih dari satu sequence diagram. Gambar di bawah ini menunjukkan ada dua interaksi yang terjadi, yaitu List client campaign dan Get campaign budget. Keyword ref menunjukkan interaksi tersebut dan mengacu ke sequence diagram List client campaign dan Get campaign budget.

Using Interaction Fragments

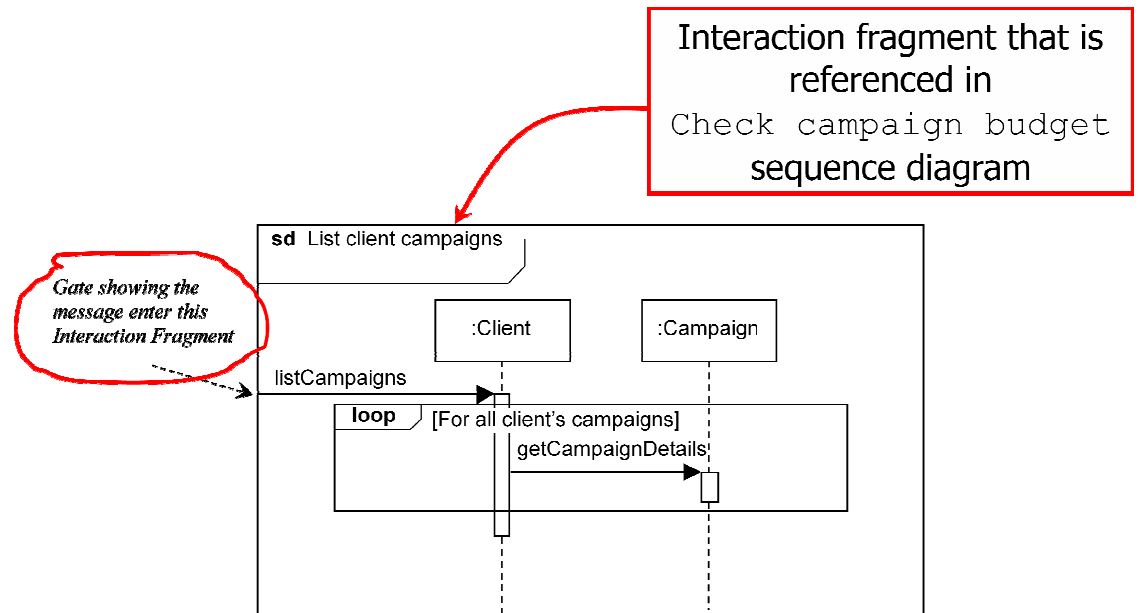


© Bennett, McRobb and Farmer 2005

24

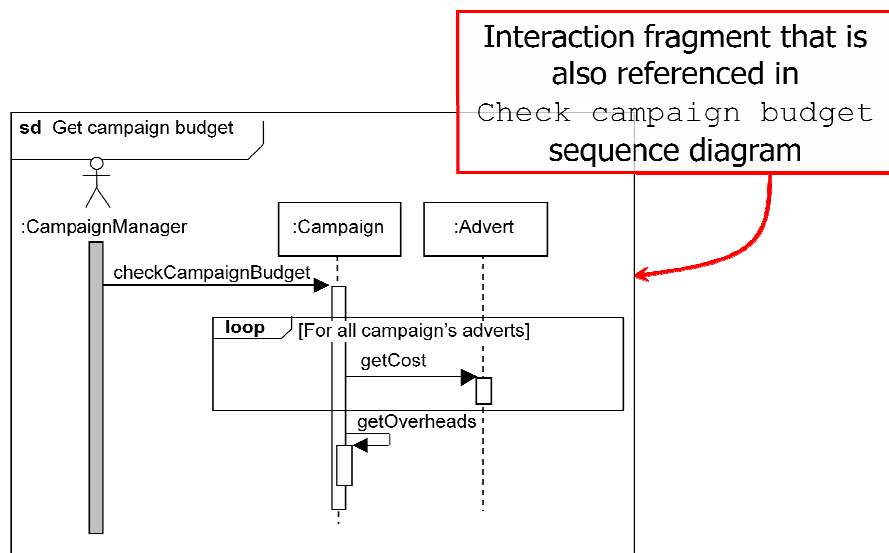
Gambar 8.12 Sequence Diagram Check campaign budget dengan interaction occurrences

Interaction Fragment



Gambar 8.13 Sequence Diagram untuk interaction fragment List client campaign

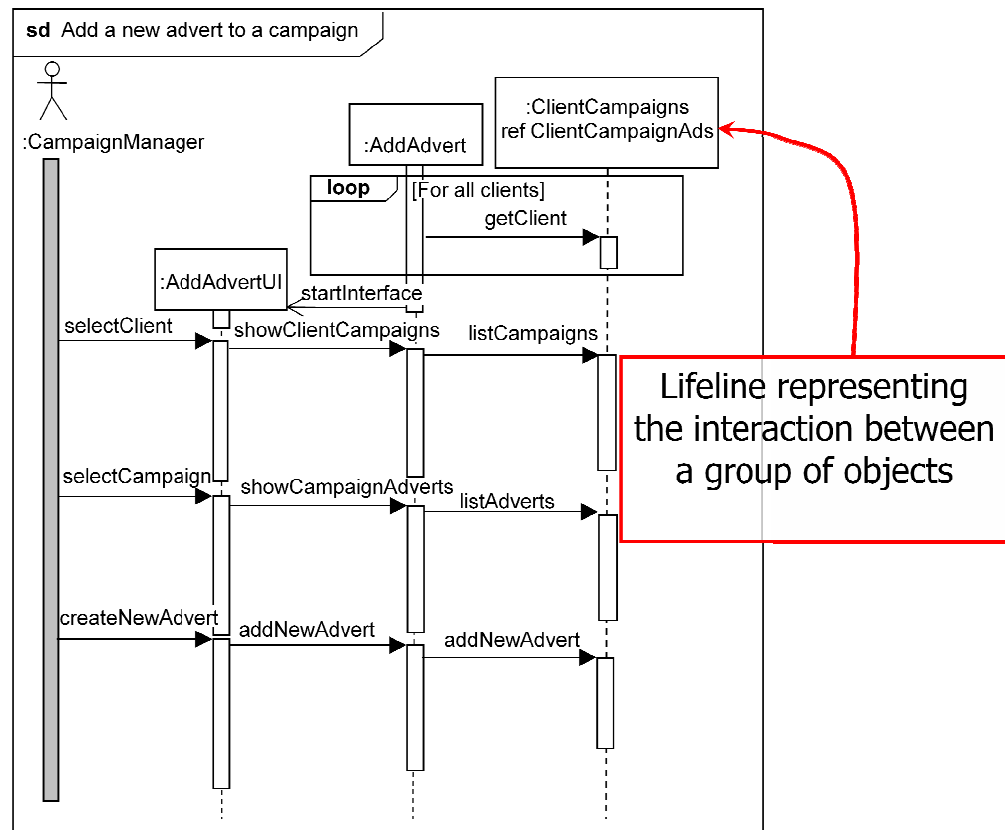
Interaction Fragment



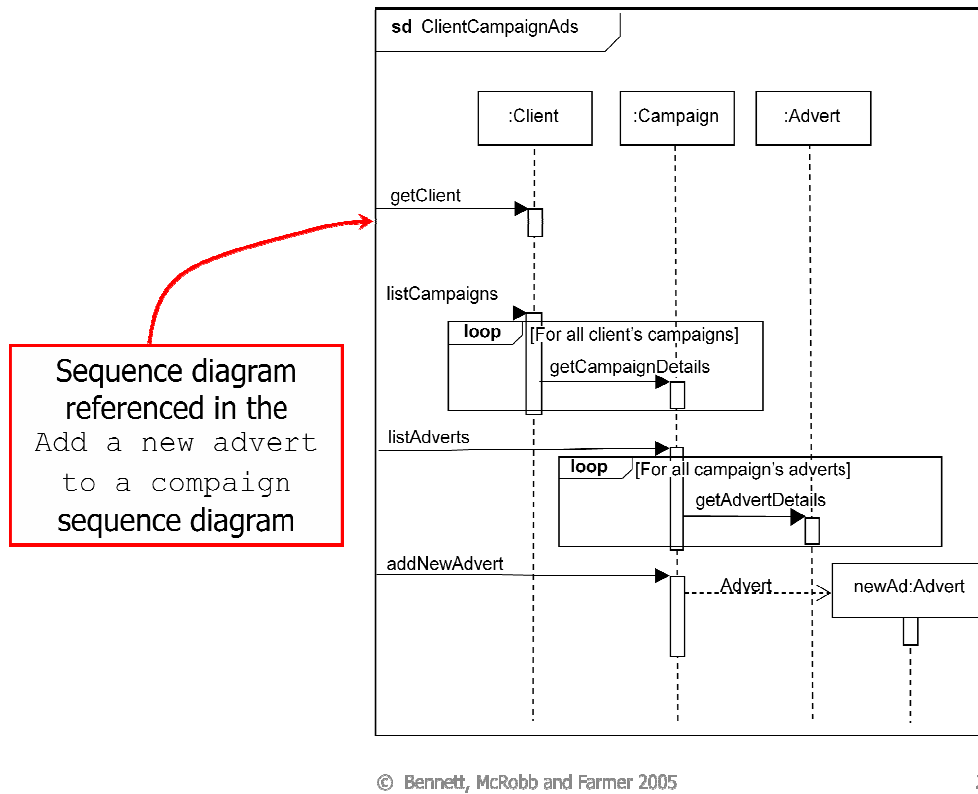
Gambar 8.14 Sequence Diagram untuk interaction fragment Get campaign budget

Lifelines untuk subsystems atau kumpulan Object

Untuk interaksi yang kompleks kita dapat menggambarkan dengan memisahkannya menggunakan lifeline untuk mewakili sekumpulan obyek dan interaksinya atau mewakili subsystem. Seperti yang digambarkan oleh gambar 8.15 dan gambar 8.16 di bawah ini.



Gambar 8.15 Sequence Diagram untuk interaction Check campaign budget

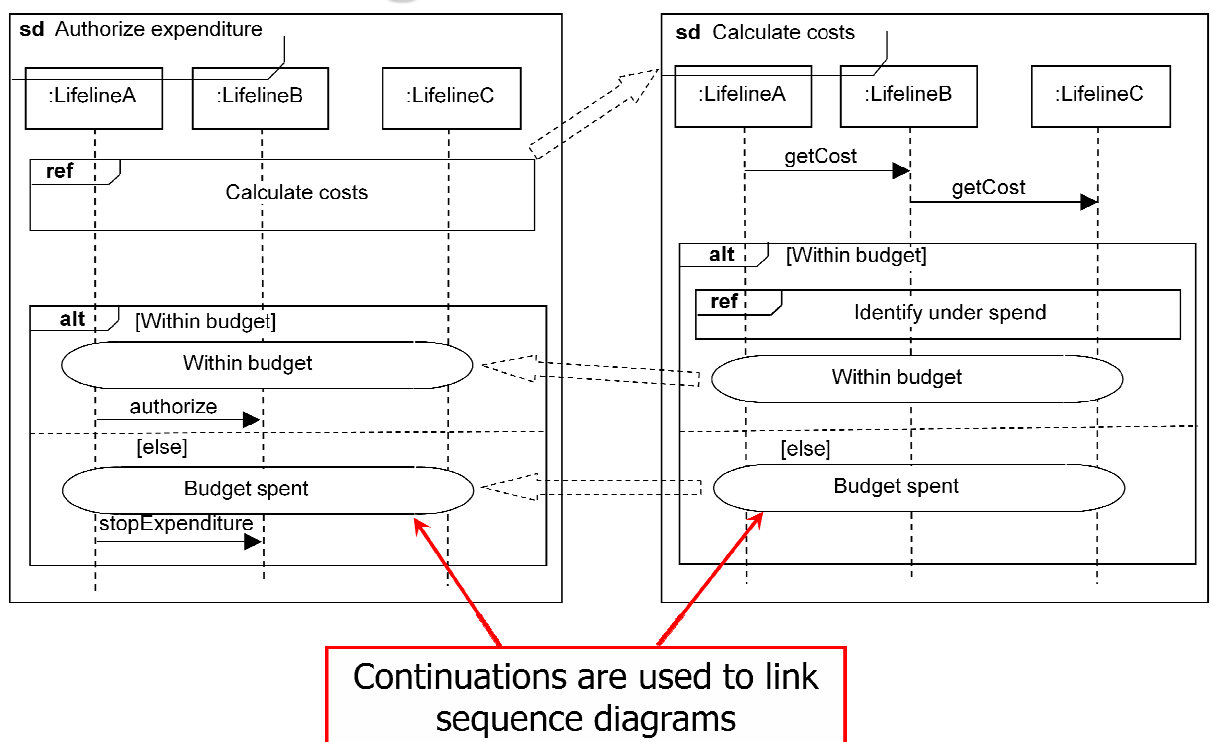


Gambar 8.16 Sequence Diagram untuk interaction fragment List client campaign

Continuations

UML 2.0 menyediakan notasi untuk menghubungkan sequence diagram. *Continuations* dapat ditentukan dengan menggabungkan fragment *alt* untuk menghubungkan kembali ke sequence diagram yang dirujuk. Gambar 8.17 menunjukkan hal tersebut.

Using Continuations



Gambar 8.17 Sequence Diagram yang menunjukkan penggunaan Continuations

8.4 Petunjuk pembuatan Sequence Diagram

Untuk pembuatan sequence diagram dapat dilakukan cara sebagai berikut :

1. Tentukan pada tingkat mana pemodelan interaksi.
2. Mengidentifikasi unsur-unsur utama yang terlibat dalam interaksi.
3. Pertimbangkan alternatif skenario yang mungkin diperlukan
4. Identifikasi interaksi yang sudah dibuat sebelumnya.

5. Gambarkan outline struktur diagram .
6. Tambahkan interaksi rinci.
7. Periksa konsistensi sesuai dengan urutan diagram
8. Periksa konsistensi dengan diagram UML lain atau model.

Referensi

1. Simon Bennet, Steve McRobb and Ray Farmer, *Object Oriented Systems Analysis and Design Using UML*, Edisi 3. ; McGraw Hill, 2006. (SB)