

USE CASE MODELLING

Pendahuluan

- Use Case adalah sebuah teknik pemodelan yang digunakan untuk menjelaskan apa yang harus dilakukan sebuah sistem baru.
- Model use case dibangun melalui sebuah proses iterasi selama diskusi antara pengembang sistem dan customer (dan / atau *end user*) berdasarkan sebuah kebutuhan khusus yang disetujui semuanya.
- Komponen utama model use case adalah *use case, actor, dan model sistem*.

Kegunaan utama use case

1. Untuk memutuskan dan menjelaskan kebutuhan fungsional sistem, menghasilkan sebuah persetujuan diantara customer (dan/ atau end user) dan pengembang software yang membangun sistem.
2. Untuk memberikan kejelasan dan penjelasan yang konsisten dari apa yang harus dilakukan sistem, sehingga selama proses pengembangan model digunakan untuk berkomunikasi ke semua personil dan menyediakan dasar untuk perancangan model.

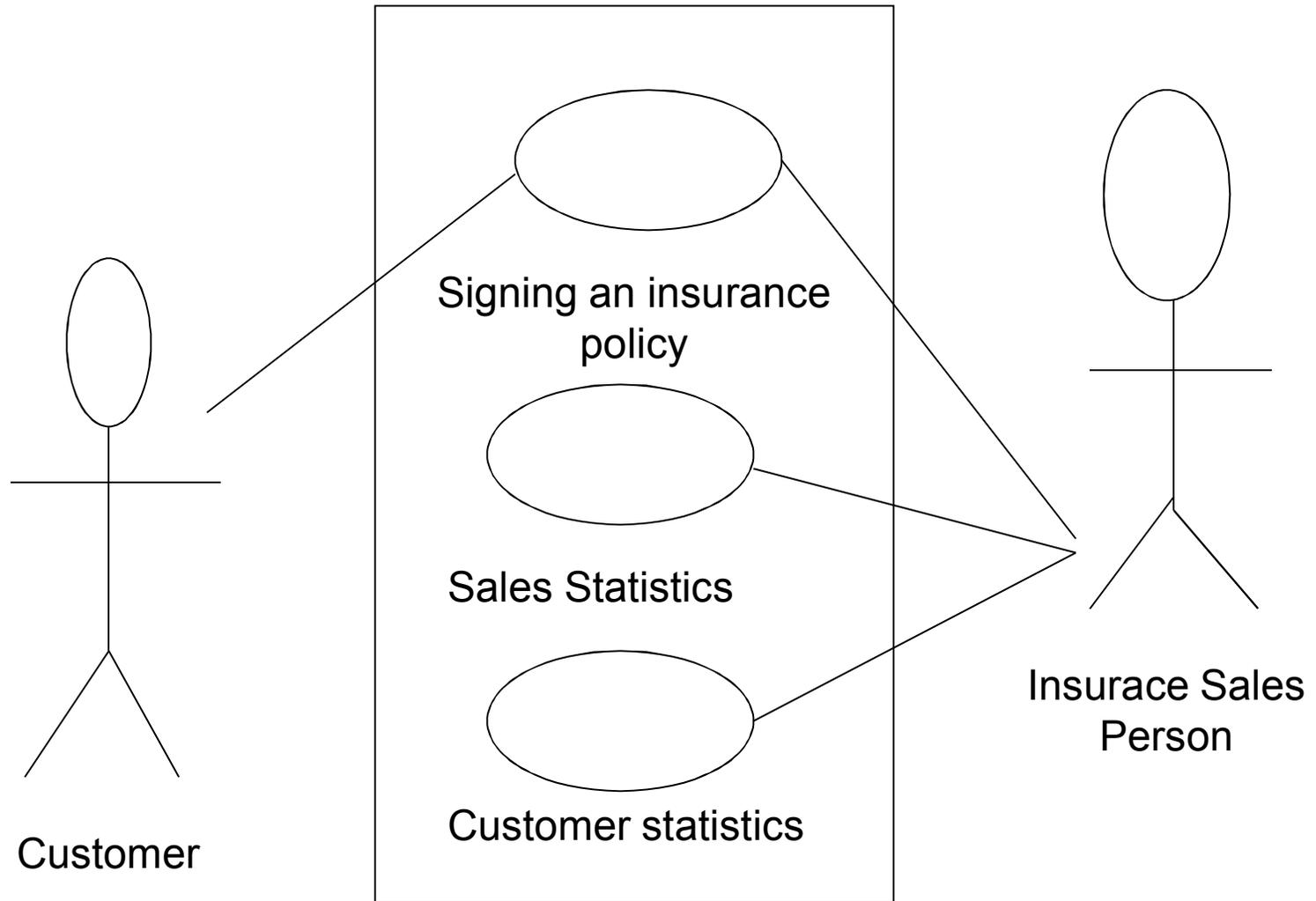
Kegunaan utama use case

3. Untuk menyediakan sebuah dasar dalam membentuk pengujian sistem yang memferifikasi sistem. Sebagai contoh, menanyakan, apakah sistem yang telah terbentuk memiliki fungsionalitas seperti yang diinginkan.
4. Untuk menyediakan kemampuan melakukan pelacakan kebutuhan fungsional kedalam kelas aktual dan operasi dalam sistem. Untuk menyederhanakan perubahan dan perluasan sistem dengan merubah model use case dan kemudian melacak efektivitas use case kedalam perancangan sistem dan implementasi.

Diagram Use Case

- Sebuah model use case dalam UML dinamakan diagram use case, dan sebuah model use case dapat dibagi kedalam sejumlah diagram use case.
- Sebuah diagram use case berisi elemen-elemen model untuk sistem, aktor, dan use case, serta memperlihatkan perbedaan relasi seperti asosiasi, generalisasi, dan ketergantungan diantara elemen-elemen. (Lihat Gambar 1)

Gambar 1. Diagram Use Case



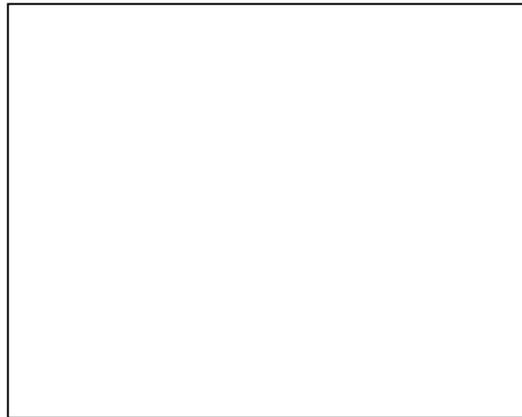
System

- Bagian dari pemodelan use case, adalah mendefinisikan dan membangun batasan (boundary) sistem.
- Sebagai catatan, yang dimaksud sistem tidak hanya sistem software, tetapi dapat berupa sebuah bisnis (pekerjaan) atau sebuah mesin.
- Mendefinisikan batasan sistem tidaklah mudah, sebab tidak dapat dilakukan secara otomatis tetapi harus dilakukan secara manual

System

- Sebuah sistem dalam diagram use case digambarkan sebagai sebuah kotak (box), nama sistem diletakkan di atas atau didalam kotak.

Insurance Business



Aktor

- Aktor adalah seseorang atau sesuatu yang berinteraksi dengan sistem; yaitu siapa atau apa yang menggunakan sistem.
- Yang dimaksud berinteraksi dengan sistem, artinya aktor mengirim dan menerima pesan ke dan dari sistem, atau mempertukarkan informasi dengan sistem.
- Singkatnya, aktor melaksanakan use case
- Aktor adalah Kelas bukan instan. Aktor merepresentasikan peran, bukan user individual dalam sistem.

Aktor

- Aktor memiliki sebuah nama, dan nama tersebut harus merefleksikan peran aktor.
- Aktor dapat dirangking. Aktor primer adalah pengguna sistem utama yaitu fungsionalitas utama.
- Aktor sekunder menggunakan fungsi sekunder dari sistem, seperti perawatan sistem, manajemen database, komunikasi, backup, dan tugas administrasi lain.
- Aktor juga dapat didefinisikan sebagai aktif dan pasif. Aktor aktif memulai use case sedang aktor pasif berpartisipasi dalam satu atau lebih use case

Finding Actor

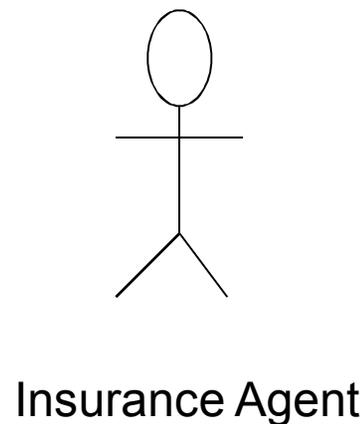
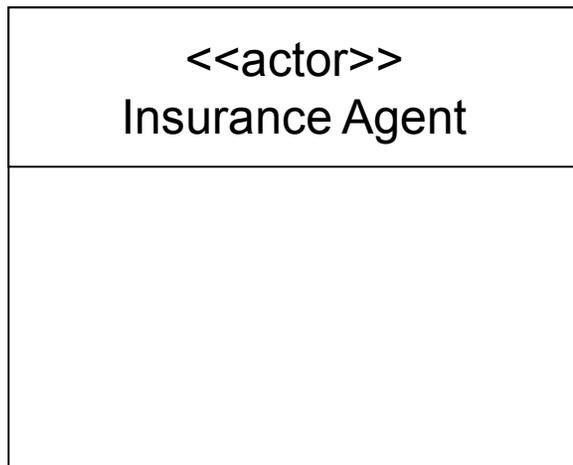
- Dengan identifikasi aktor, kita yakin bahwa entitas-entitas tersebut tertarik menggunakan dan berinteraksi dengan sistem.
- Aktor dapat diidentifikasi dengan menjawab pertanyaan berikut :
 1. Siapa yang akan menggunakan fungsi utama sistem (aktor primer) ?
 2. Siapa yang akan memerlukan bantuan dari sistem untuk mengerjakan tugas hariannya ?
 3. Siapa yang akan memerlukan perawatan, administrasi, dan menjaga sistem bekerja (aktor sekunder) ?

Finding Actor

4. Perangkat hardware mana yang perlu ditangani sistem ?
5. Dengan sistem lain yang mana, sistem perlu berinteraksi ? Sistem memasukkan komputer lain sebaik aplikasi lain dalam komputer dimana sistem akan beroperasi.
6. Siapa atau apa yang tertarik dengan hasil (nilai) yang dihasilkan sistem ?

Aktor didalam UML

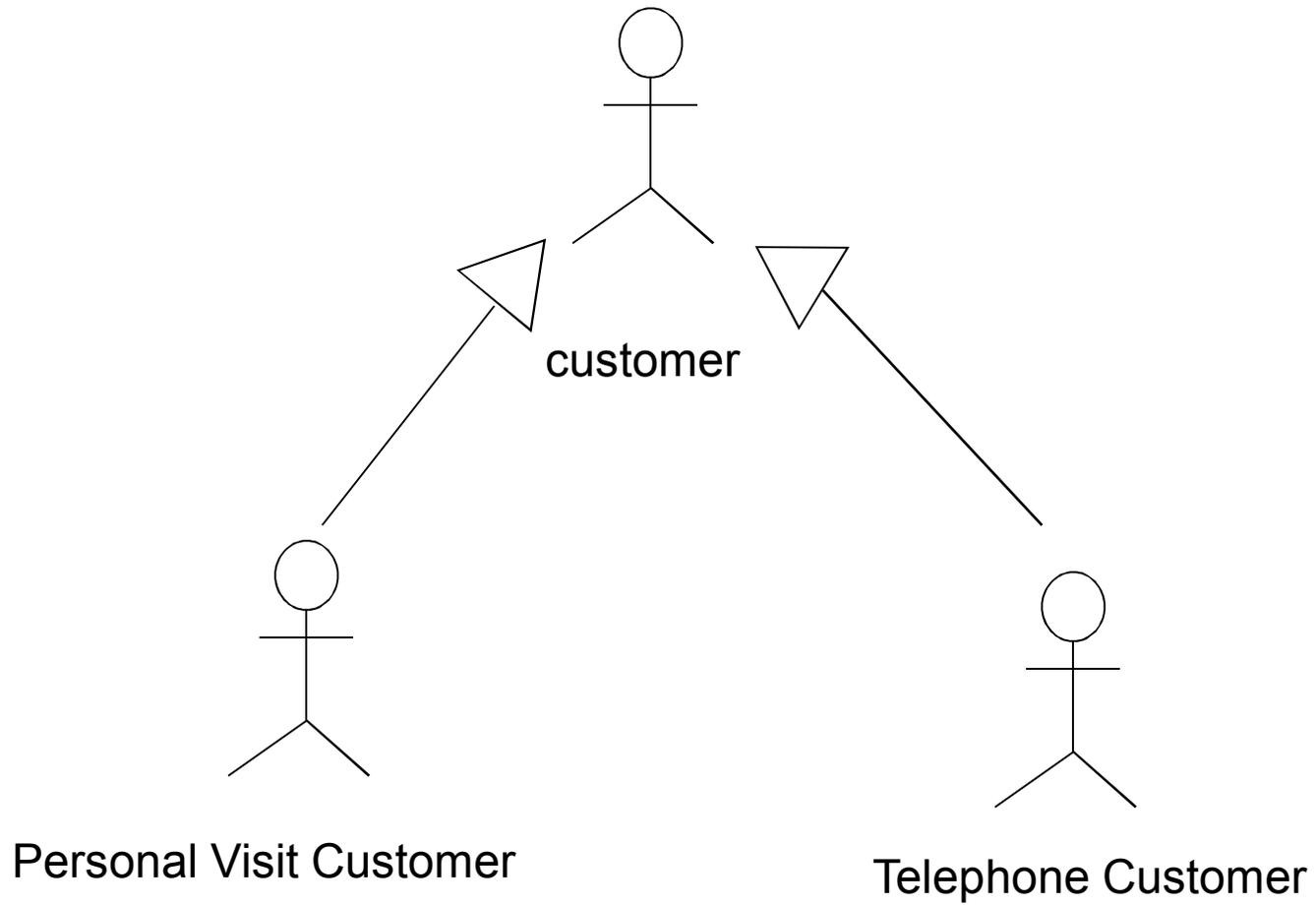
- Aktor adalah sebuah Kelas yang diperlihatkan sebagai sebuah kotak dengan stereotype <<actor>>.
- Stereotype icon dalam use case digambarkan sebagai manusia kawat yang memiliki nama dibawah gambar.



Relasi diantara Aktor

- Karena aktor adalah sebuah kelas, maka aktor memiliki relasi seperti relasi yang dimiliki oleh kelas.
- Dalam diagram use case, hanya relasi generalisasi yang digunakan untuk menjelaskan tingkah laku umum diantara sejumlah aktor.
- Ketika ada beberapa aktor, sebagai bagian dari peran mereka, juga memainkan peran yang lebih general, yang didiskripsikan sebagai sebuah generalisasi. Lihat gambar berikut.

Generalisasi



Use Case

- Use case merepresentasikan fungsi lengkap yang dirasakan oleh aktor.
- Use case dalam UML didefinisikan sebagai rangkaian kegiatan yang membentuk sebuah sistem yang dihasilkan dari hasil observasi nilai ke aktor tertentu.
- Kegiatan-kegiatan dapat memasukkan komunikasi dengan sejumlah aktor (user dan sistem lain) sebaik kalkulasi performa dan pekerjaan di dalam sistem.

Karakteristik use case

- Sebuah use case selalu diawali oleh sebuah aktor. Use Case dibuat berdasarkan kepentingan aktor. Aktor harus secara langsung atau tidak langsung memerintah sistem untuk membentuk use case.
- Sebuah use case menyediakan nilai ke aktor. Use case harus mengirimkan beberapa nilai yang nyata ke user.

Karakteristik use case

- Sebuah use case adalah lengkap. Use case harus memiliki deskripsi yang lengkap. Kesalahan yang sering dilakukan adalah membagi use case ke dalam use case yang lebih kecil yang mengimplementasikan setiap perintah sebagai sebuah fungsi yang dipanggil dari bahasa pemrograman lain. Sebuah use case tidak lengkap hingga nilai akhir dihasilkan, meskipun beberapa komunikasi (seperti user dialog) terjadi sepanjang waktu

Finding Use Case

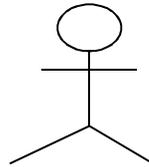
- Aktor membutuhkan fungsi yang mana dari sistem?
- Apakah aktor memerlukan *read, create, destroy, modify, or store* beberapa informasi dari sistem?
- Apakah aktor memiliki catatan tentang peristiwa dalam sistem, atau apakah aktor memerlukan catatan sistem tentang sesuatu? Apakah peristiwa tersebut menunjukkan fungsi sistem?
- Dapatkah pekerjaan harian aktor disederhanakan atau dibuat lebih efisien melalui fungsi sistem yang baru?

Finding Use Case

- Input/Output apa yang diperlukan sistem?
Darimana/kemana input/output tersebut?
- Apa masalah utama pada sistem utama sistem? (Barangkali sistem manual dapat diganti dengan sistem otomatis)

Use case dalam UML

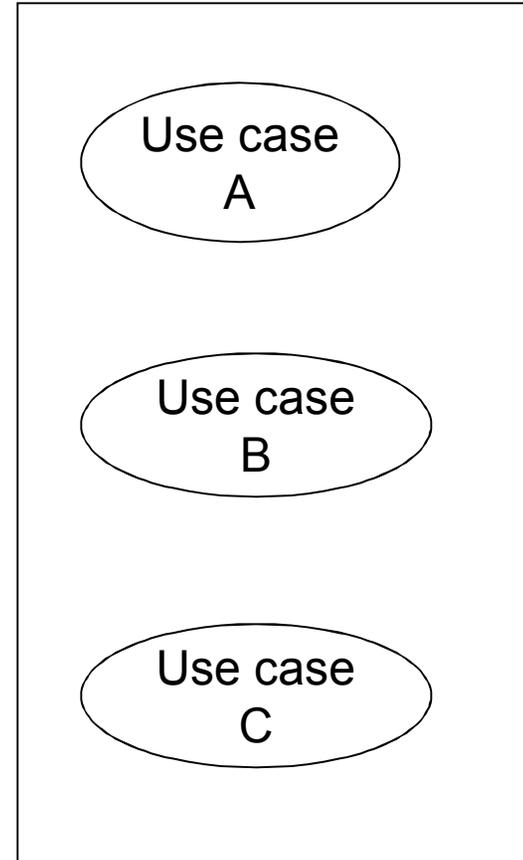
Communication
Association



Actor Name

Use-case
Name

System Name



Relasi diantara Use case

1. Relasi Memperluas (extends): Relasi generalisasi dimana sebuah use case memperluas use case lain dengan menambah kegiatan ke sebuah use case general. Perluasan use case termasuk tingkah laku dari use case diperluas, tergantung pada kondisi perluasan.
2. Relasi Penggunaan (uses): Relasi generalisasi dimana sebuah use case menggunakan use case lain, diindikasikan sebagai bagian dari spesialisasi use case, tingkah laku dari use case general juga termasuk.

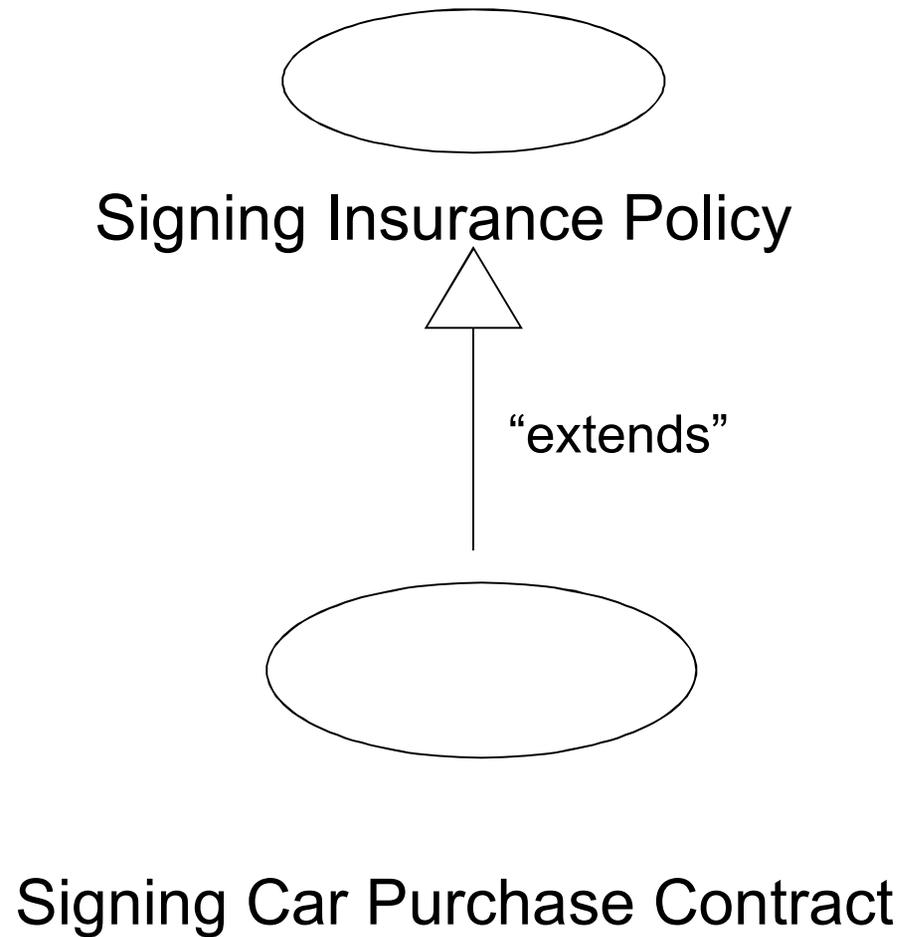
Relasi diantara Use case

- Pengelompokan (group): Ketika sejumlah use case memiliki fungsi yang hampir sama atau beberapa memiliki relasi satu dengan lainnya, mereka dapat dibundel dalam sebuah paket. Sebuah paket merelasikan elemen-elemen model, dan dapat diperluas atau dipersempit kedalam icon, tergantung bagaimana developer memandang sebuah paket pada satu waktu. Paket tidak memiliki arti semantik lain.

Relasi Perluasan (extends)

- Ketika sebuah use case memperluas use case lain, berarti use case tersebut memasukkan beberapa tingkah laku dari use case untuk diperluas.
- Sebuah use case yang telah diperluas, dapat menangani pengecualian (exception) yang merupakan kasus khusus dari sebuah use case.
- Relasi perluasan diantara 2 use case dinyatakan dengan relasi generalisasi (sebuah garis dengan ujung segitiga) dengan stereotype <<extends>>.

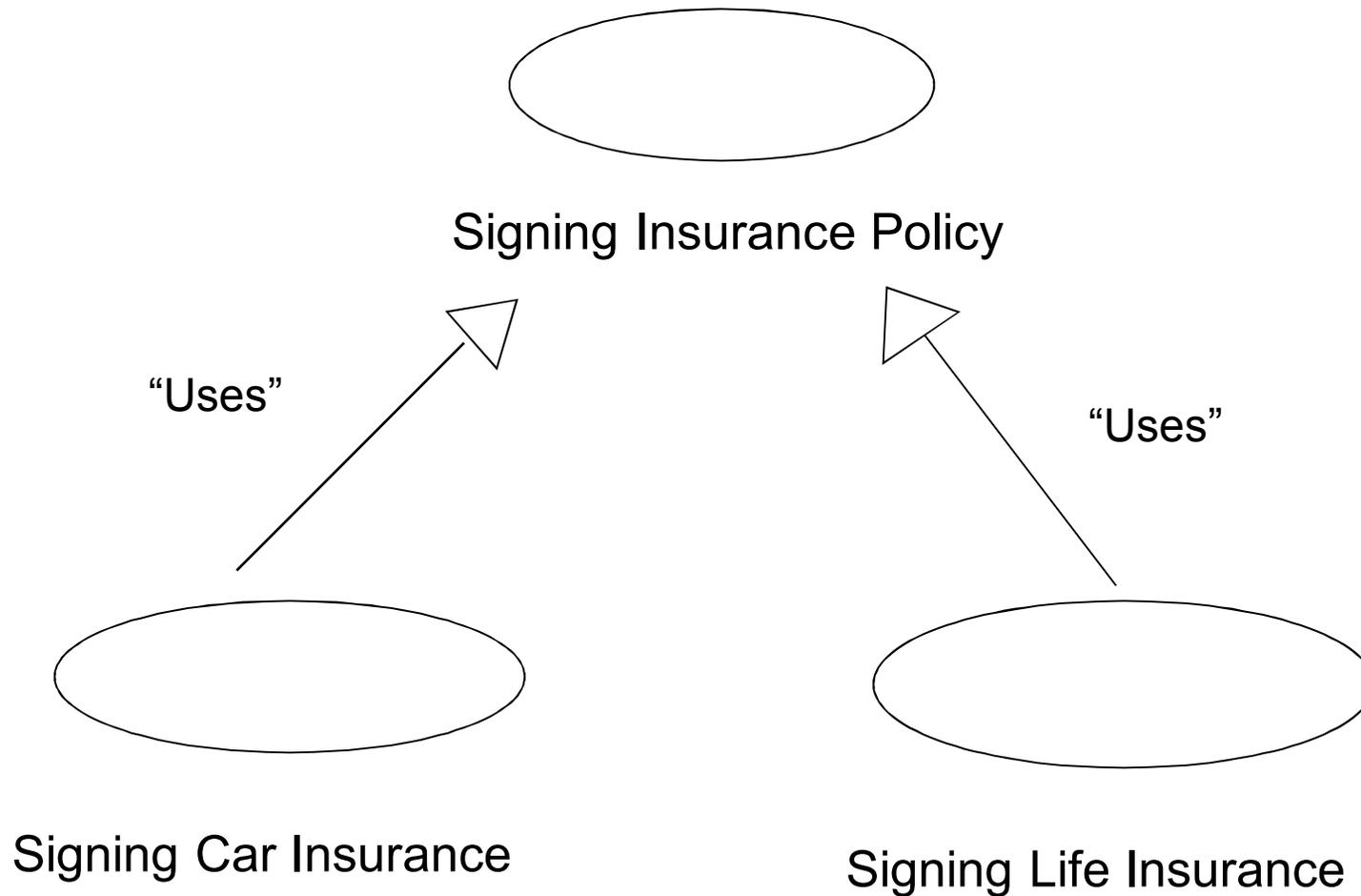
Relasi Perluasan (extends)



Relasi Penggunaan (uses)

- Ketika sejumlah use case memiliki kesamaan tingkah laku, tingkah laku tersebut dapat dimodelkan dalam use case tunggal yang digunakan oleh use case lain.
- Ketika sebuah use case menggunakan use case lain, seluruh use case harus digunakan (melalui aktifitas dalam use case)
- Jika use case yang digunakan tidak pernah digunakan sendiri, use case tersebut dinamakan *abstract use case*.
- Relasi menggunakan mempergunakan relasi generalisasi.

Relasi Penggunaan (uses)

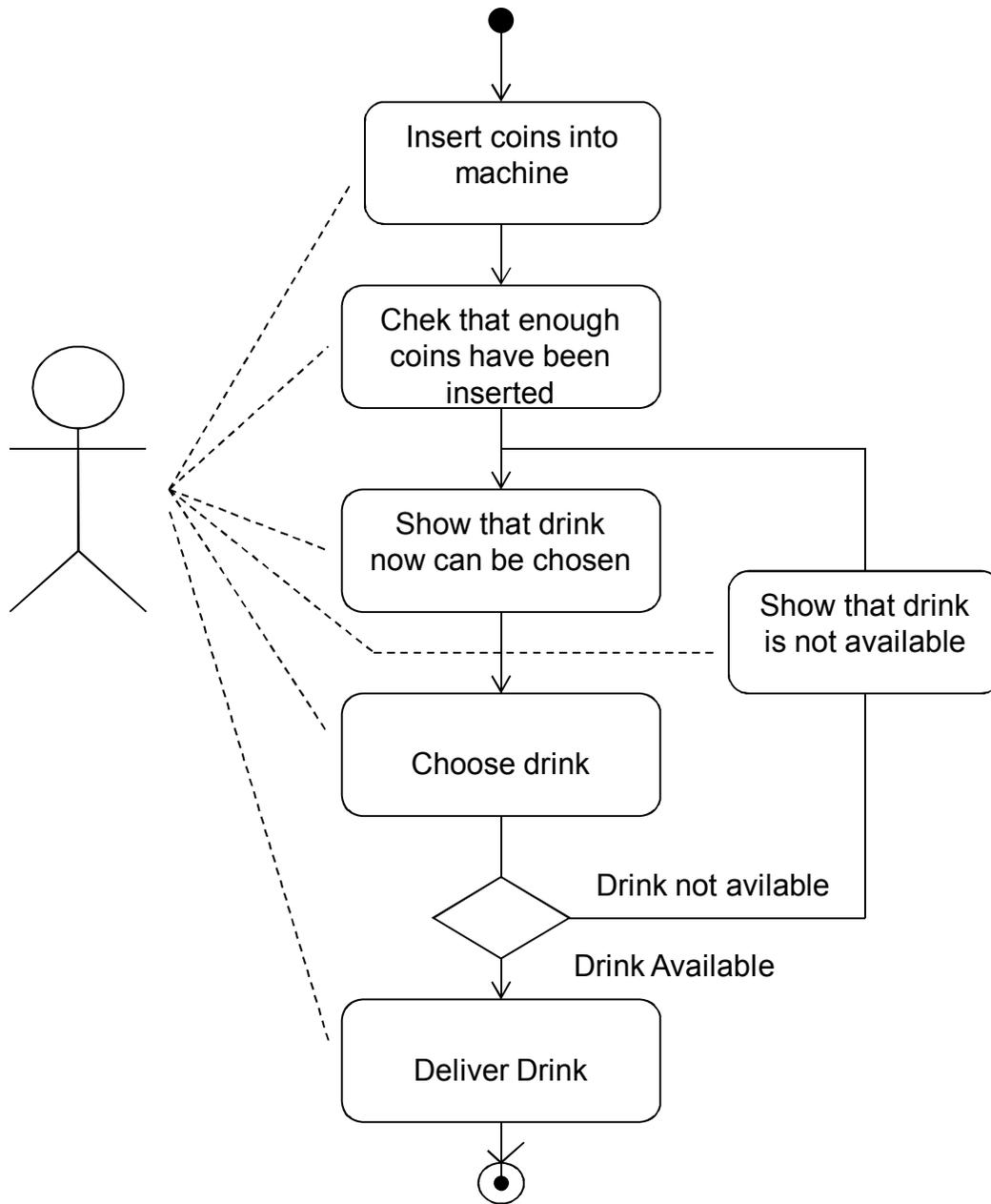


Penggambaran Use Case

- Menjelaskan bagaimana aktor dan use case saling berinteraksi, dengan menjelaskan :
 1. Tujuan use case.
 2. Bagaimana use case diinisialisasi
 3. Aliran pesan antara aktor dan use case
 4. Bagaimana use case menyelesaikan sebuah nilai ke aktor

Penggambaran Use Case

- Use case juga dapat digambarkan dalam bentuk diagram aktifitas (activity diagram) seperti dalam gambar slide berikut.
- Diagram aktifitas memperlihatkan rangkaian aktifitas, urutannya, dan pilihan keputusan yang dibuat untuk mengindikasikan aktifitas mana yang akan dilakukan selanjutnya.
- Model use case harus mudah dikomunikasikan ke user dan tidak harus digambarkan secara formal.



Penggambaran Use Case

- Penggambaran use case sebagai sebuah komplemen berisi diskripsi lengkap dan umum, sejumlah skenario aktual yang digunakan untuk mengilustrasikan apa yang terjadi ketika sebuah use case di-instansiasi.
- Skenario mengilustrasikan sebuah kasus khusus, dengan aktor dan use case sebagai instan aktual.
- Customer dapat memahami sebuah use case yang kompleks ketika skenario yang lebih praktis menjelaskan tingkah laku sistem.

Pengujian Use Case

- Pengujian use case melalui tahap verifikasi dan validasi.
- Verifikasi berarti mengkonfirmasi sistem telah dibangun secara benar atau menurut spesifikasi pembuatan.
- Validasi menjamin sistem yang sedang dibangun benar-benar yang dibutuhkan customer atau end user.

Menjalankan Use Case

- Sebuah teknik yang bagus yang diimplementasikan selama definisi dan pengujian use case disebut menjalankan use case (walking the use case).
- Dalam teknik ini, orang-orang yang berbeda dalam kelompok model memainkan peran aktor dan sistem dalam penggunaan khusus.
- Orang memainkan peran seorang aktor memulai dengan mengatakan apa yang dibutuhkan aktor dengan sistem.

Menjalankan Use Case

- Hasil eksekusi sistem dimulai dari kegiatan ini, orang yang berperan dalam sistem kemudian mengatakan apa yang dia lakukan ketika use case dieksekusi ?
- Developer tidak termasuk peran yang dimainkan.
- Dengan cara ini diketahui kegiatan apa yang perlu didiskripsikan dan apa yang tidak perlu didiskripsikan.
- Tiap-tiap orang dapat bertukar peran untuk pengujian yang lebih baik.