Design Principles Behind Smalltalk

Principios de diseño de Smalltalk:

Se concentra en dos áreas principales de investigación:

- Un lenguaje de descripción (lenguaje de programación) que sirve de interfaz entre los modelos en la mente humana.
- Un lenguaje de interacción (User interface) que conecta la interacción humana.

Dominio personal: Si un sistema es para servir al espíritu creativo, debe ser completamente entendible para un individuo solitario. "Cualquier barrera que exista entre el usuario y alguna parte del sistema será finalmente una barrera a la expresión creativa".

Buen Diseño: Un Sistema debería ser construido con un mínimo conjunto de partes no modificables, esas partes deberían ser tan generales como sea posible.

Lenguaje:

La comunicación entre dos personas (o una persona y una computadora) incluye una comunicación en dos niveles. Explicita: Incluye la información que es transmitida en determinado mensaje. Implícita: Incluye las suposiciones relevantes comunes a los dos seres.

Propósito del lenguaje: Proveer un esquema para la comunicación.

Alcance: El diseño de un lenguaje para usar computadoras debe tratar con modelos internos, medios externos, y la interacción entre ellos.

Objetos que se comunican:

Objetos: Un lenguaje de computación debe soportar el concepto de "objeto" y proveer una manera uniforme de referirse a los objetos del universo.

Administración del almacenamiento: Para ser auténticamente "orientado a objetos" un sistema debe proveer administración automática del almacenamiento.

Mensajes: La computación debería ser vista como una capacidad intrínseca de los objetos que pueden ser involucrados uniformemente enviándoles mensajes.

Metáfora uniforme: Un lenguaje debería ser diseñado alrededor de una metáfora poderosa que pueda ser aplicada uniformemente en todas áreas.

Organización:

Al incrementarse la cantidad de componentes de un sistema, la probabilidad de interacción no deseadas crece rápidamente.

Modularidad: Ningún componente en un sistema complejo debería depender de los detalles internos de ningún otro componente.

Clasificación: Un lenguaje debe proveer un medio para objetos similares, y para agregar nuevas clases de objetos en pie de seguridad con las clases centrales del sistema.

Polimorfismo: Un programa solo debería especificar el comportamiento esperado de los objetos, no su representación.

Factorización: Cada componente independiente de un sistema solo debería aparecer en un solo lugar. Esto le da un buen reaprovechamiento

Máquina virtual: Una especificación de máquina virtual establece un marco para la aplicación de tecnología.

Interfaz de usuario:

Un lenguaje en el que la mayor parte de la comunicación es visual.

Principio reactivo: Cada componente accesible al usuario debería ser capaz de presentarse de una manera entendible para ser observado y manipulado.

Sistema Operativo: Un sistema operativo es una colección de cosas que no encajan dentro de un lenguaje. Smalltalk no tiene "sistema operativo" como tal. Las operaciones primitivas necesarias, como leer una página del disco, son incorporadas como métodos primitivos.