

# A Summary of pilot projects on XVCL at SES Systems Pte Ltd



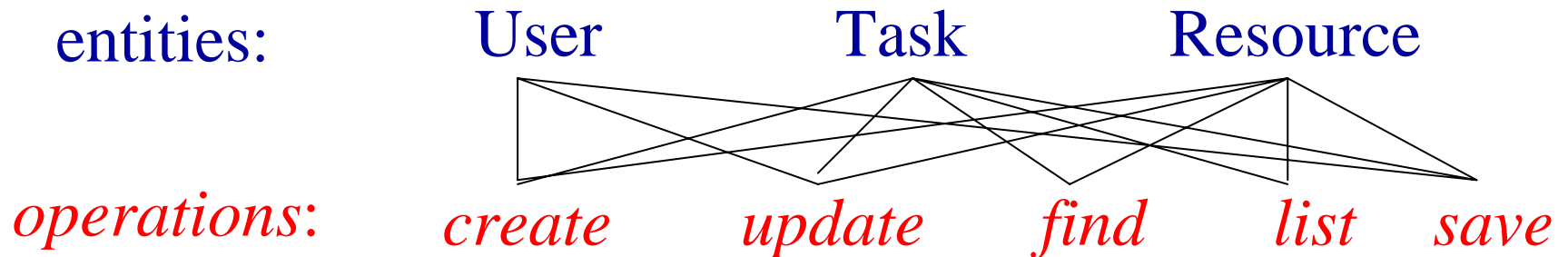
# Pilot project 1:

- Command and control (C2) application
- Developed in C# on .NET platform
- Objective:
  - to enhance maintainability and facilitate reuse
- Method:
  - we identified similar patterns in the application that led to redundancies in the OO solution
  - we created generic meta-program structures in XVCL to eliminate redundancies and facilitate reuse

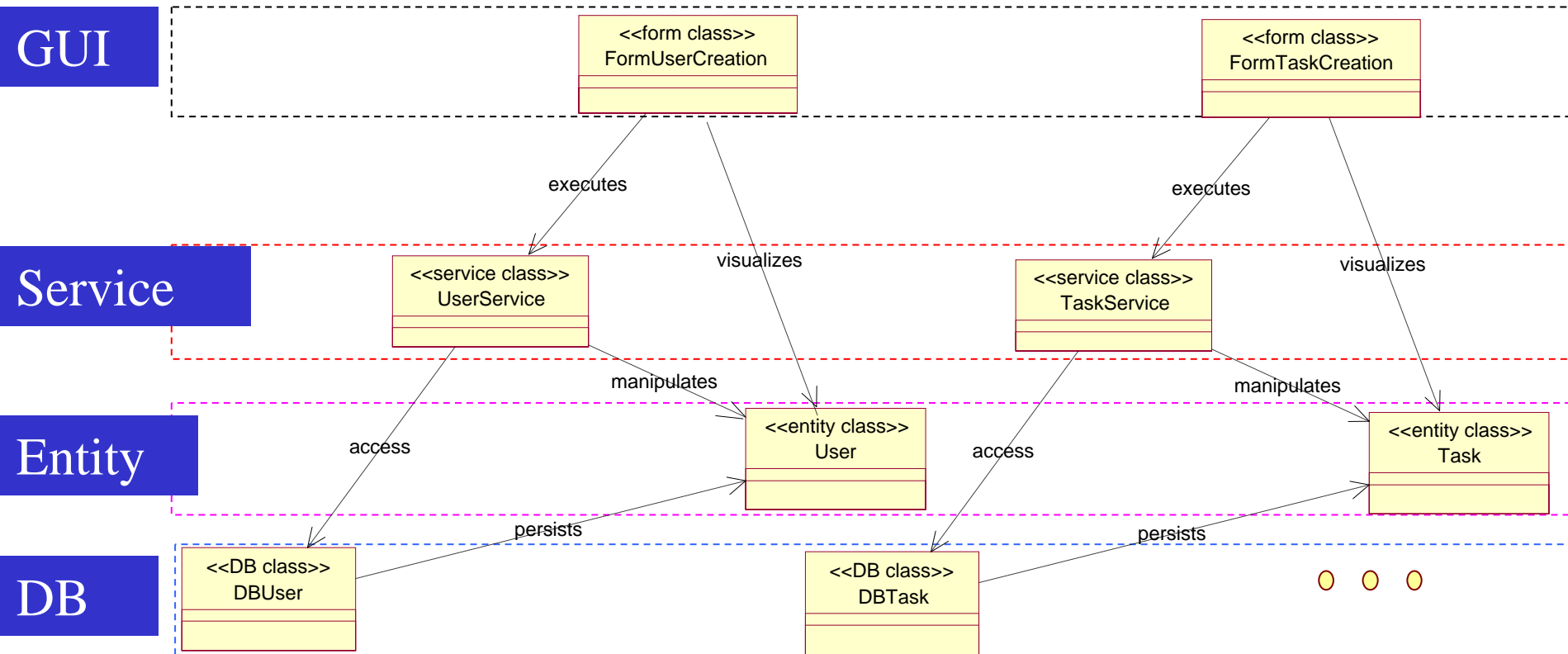
- The scope of the C2 application:
  - 13 domain entities (User, Task, Resource, etc.)
  - operations: create, update, find, etc. for each entity
  - 117 classes in GUI, service / entity / DB layers
- Results:
  - we eliminated 68% of code at the meta-level
  - estimated 8-fold reduction of the maintenance effort

# The source of redundancies

- each entity (User, Task) needs similar operations:



- operations (*create, update, etc.*) follow the same pattern independently of the entity



# Summary of the classes

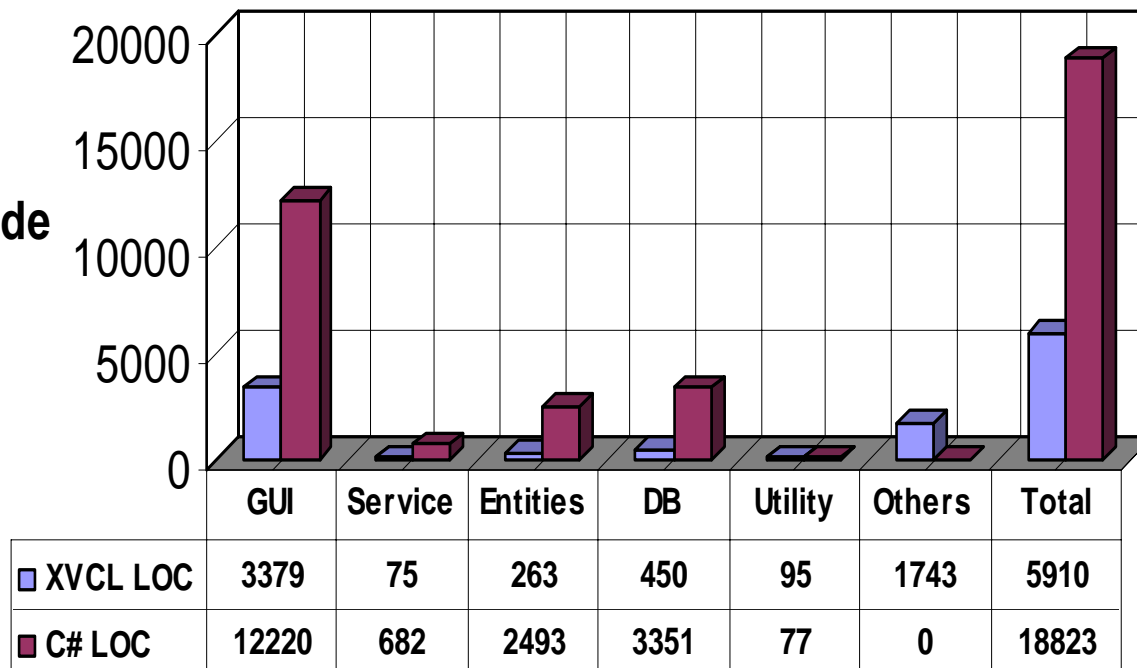
- 117 classes in GUI, service, entity, DB access

Layer	Examples	# classes
GUI	forms, controls, buttons	59
Service	Create, Delete, Update	19
Entity	Task, User, Resource	26 (counting both entity and entity list classes)
DB	database access classes	13

# Lines of code comparison

## Lines of code comparison between XVCL and C#

Lines of code (LOC)



■ XVCL LOC  
■ C# LOC

Components

# Pilot 1: Experience, Results

- XVCL was easier to apply than expected
- we got encouraging results
- estimated maintenance effort:
  - to add a new domain entity to the application:

with XVCL (avg)	without XVCL (avg)
133 LOC	1440 LOC
2 man/hours	16 man/hours

# Pilot project 2:

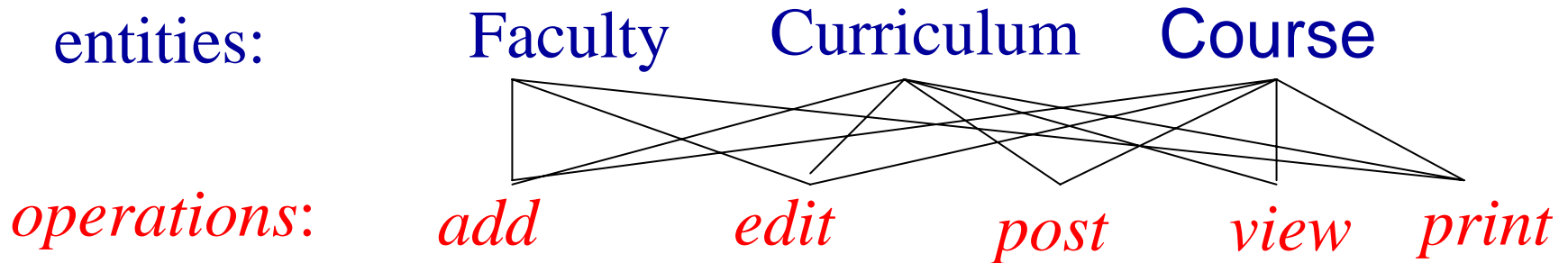
- Web Portal System for e-learning:
  - Information management system
  - based on existing ASP-based portal framework
- Objective:
  - to enhance maintainability and facilitate reuse
  - to facilitate rapid development
- Method:
  - we represented generic entities and their relationships with XVCL
  - we defined methods and model-based development process of deriving concrete web-systems using XVCL.
  - we created meta-program structures that facilitate generation of similar information management web-portals

# The project scope and results

- The scope:
  - 20 new domain entities
  - relationships between entities (aggregations, association, abstract inheritance, etc.)
  - Operations: access controls, view, edit, create, upload, search, etc.
- Results:
  - eliminated ~80% of code at the meta-level for 20 domain entities
  - project specific XVCL code: 4.9k lines
  - generated project specific code: 25k lines (ASP code)

# The source of redundancies

- each entity (faculty, curriculum, course, etc.) need similar set of operations:

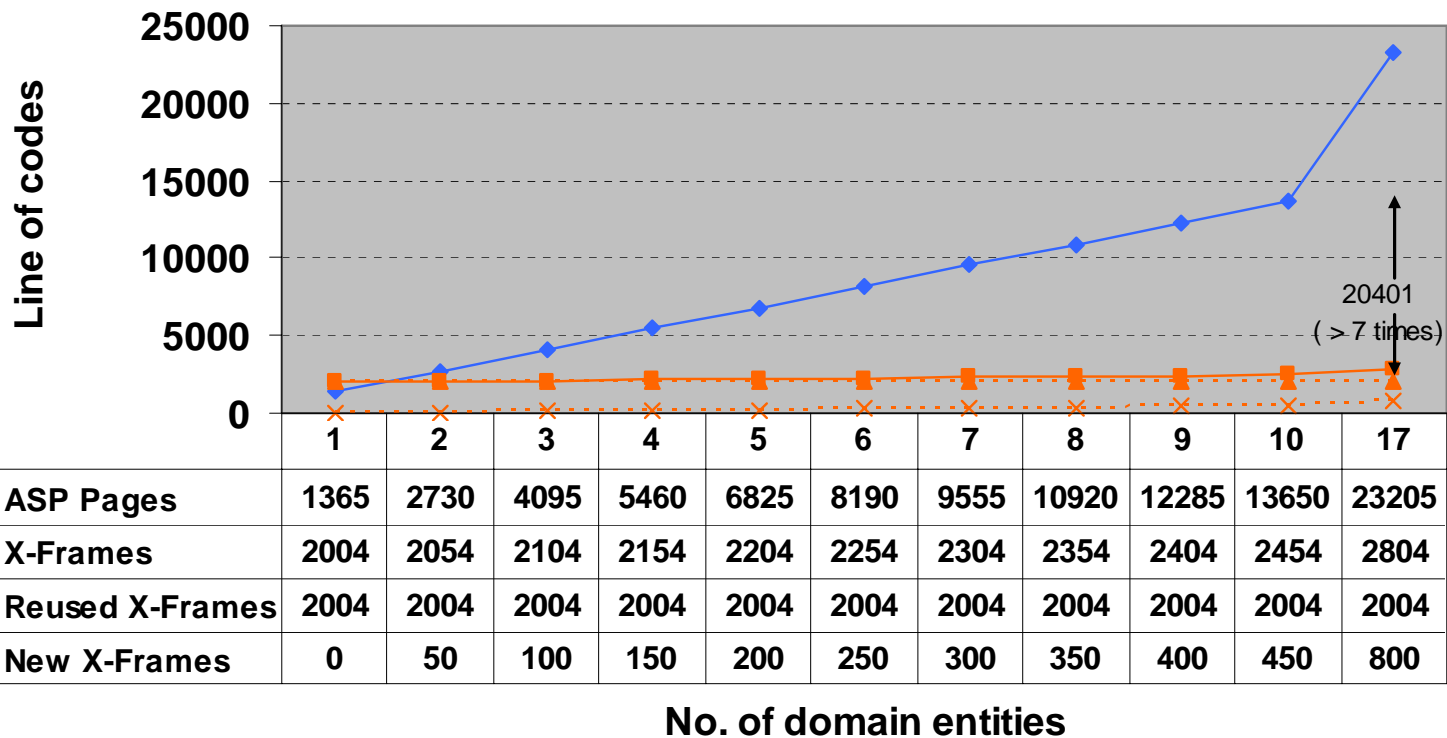


- operations (view, edit, add, print, etc.) are handled by a set of similar ASP pages independent of the entity

# ASP pages for each entity

ASP page	Description
<b>Presentation Level</b>	
<b>view.asp</b>	displays the details of an entity object
<b>edit.asp</b>	displays the details of an entity object for adding, editing and deleting an entity
<b>print.asp</b>	displays the printable view of the web page
<b>post.asp</b>	displays the list of postings for adding, editing and deleting remarks or comments related to an entity
<b>Framework Level</b>	
<b>inc_win_main.asp</b>	include file for listing of all the entity objects and provides optional search or filtering capability
<b>inc_win_mini.asp</b>	include file for listing the summary of all entity objects
<b>inc_win_view.asp</b>	include file for displaying the details of an entity object
<b>inc_win_form.asp</b>	include file for the adding, editing and deleting an entity object
<b>inc_post_page.asp</b>	include file that enable the adding, editing and deleting of remarks or comments related to the entity objects
<b>inc_DbEntity.asp</b>	Include file contains implementation of a domain entity class

## ASP codes VS XVCL codes



◆ ASP Pages 
 ■ X-Frames 
 ▲ Reused X-Frames 
 × New X-Frames

# Pilot 2: Experience, Results

- XVCL was **even** easier to apply than expected
- we got **more** encouraging results:
  - able to deliver the sample system on time
  - able to create other similar portals rapidly
  - to add a new domain entity to the web portal:

with XVCL (avg)	without XVCL (avg)
50-60 LOC	1000-1500 LOC
2-5 minutes	2-6 man/hours (excluding time for defining domain model)

- has not reached its maturity for industrial wise application:
  - short in Guidelines and Methodology
  - short in supporting tools (e.g., editors, reverse engineering, debugger, etc)
- collaborations with NUS continues
  - Guidelines and Methodology
  - publish papers
  - more experiments

- reduces the development time and effort
- reduces code redundancies by 68% - 80%
- free and open source!  
(<http://fxvcl.sourceforge.net/>)
- language is simple and easy to learn
- experimental studies gives encouraging results
- new and need to wait for its maturity