

Integrating with Existing (Including Legacy) Applications

- Why integration is important
- A framework for discussing integration
- Integration of:
 - Enterprise applications
 - B2B integration
 - Legacy app integration

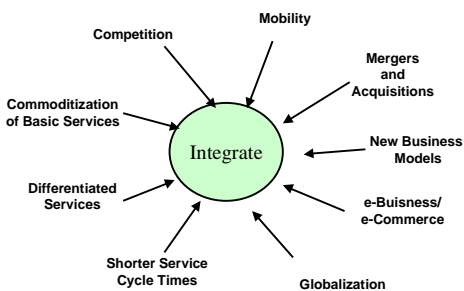
Amjad Umar

Initial Thoughts

- On the average, a customer purchase involves 10 to 11 applications
- Some integrations within large orgs involve hundreds (example: GTE's 1000+)
- B2B trade increases the number of systems involved
- The systems that need to interwork/interoperate
 - range from very new to 20+years old (legacies)
 - many components (networks, operating systems, apps, business processes) from different suppliers
 - requirements of individual systems not documented
- These systems, if not integrated, can increase service time, introduce errors, increase "hassles"
- Integration: minimizes human effort in getting the job done
- Integration is at several levels (networks to business processes)
- Enterprise Application Integration (EAI) platforms are good way of integrating apps

Copyright (A. Umar) 2002

Why Integrate: Promises and Pitfalls



Many drivers for integration, however:

- integrations are expensive
- Integrations can introduce performance and security risks

Copyright (A. Umar) 2002

Technologies to support Ebusiness

qāñē cē-āē Eoc'100 f	cēñcēāī -lācā	mēcō- āāāō	ā-ī- mēcēē	ācēñē ī-ēē	kēñī cē āē
OMM	t EAI t ē'nal s' fml atēñāic atēñāifet	t-ī HHI g-ī-ī c'cāl ācā cācā ēāte	a_ōl t-ē-āñē f' p ēcā ē l' l a_	t EAI t-ē-āñē c'j l t-ē-āñē ī cēñāc ī ēñb ā-ācēñē -ēñil	fm
IVME	atēñāic ēl-ē-ā māñāifet POTM	t-ī HHI l-ē-āñē c'cāc	a_ōl t-ē-āñē f' p ēcā ē l' l a_	t EAI t-ē-āñē ā-ācēñē -ēñī cēñ-ī cēñāc ī ēñil	fml pk'1
IVME	māñāifet POTM ēcā ē c'cācē	l c'cāl mī ml ēñēāā ē	a_ōl t-ē-āñē f' p	ēcā ē t-ē-āñē f' p ēñācēñē kēñī ēā	pk'1 āñāñāñil kēñī ēā
IVME	māñāifet POTM mī ml ēñēāā ē	c'cāl sp'j l f' p fñj p	ēcā ē t-ē-āñē f' p fñj p	pk'1 āñāñāñil	

Copyright (A. Umari) 2002

- # Integration Concepts
- Key idea: Integrate applications
 - Within the enterprise
 - External partners
 - Integration: minimizes user effort in getting the job done
 - Approaches to application integration
 - interconnect at data, program, user interface
 - Can use loose or tight coupling
 - can integrate by using point to point or hub/bus approaches
- Copyright (A. Umari) 2002

Applications Can Interconnect at data, method or user interface level

The diagram illustrates the levels of application interconnection. It is divided into three main sections: Information Consumer, Mediator, and Information Supplier.

- Information Consumer:** Contains a yellow box labeled ". Web Browser", ".Java applet", and ".Other apps".
- Mediator:** A central box labeled "Interconnection Technologies" containing three orange boxes: "Remote User Connector", "Remote Method Connector", and "Remote Data Connector".
- Information Supplier:** Contains a green oval labeled "User Interfaces", a light blue box labeled "Application Code", and a green cylinder labeled "Data Source".

Arrows indicate the flow of interaction:

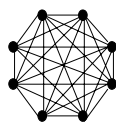
- From "User Interfaces" to "Remote User Connector".
- From "Remote User Connector" to ". Web Browser".
- From "User Interfaces" to "Application Code".
- From "Application Code" to "Remote Method Connector".
- From "Application Code" to "Data Source".
- From "Data Source" to "Remote Data Connector".
- From "Remote Data Connector" to ".Other apps".

A note at the bottom left states: "User Interface connection is the last resort -- if cannot connect at method or data".

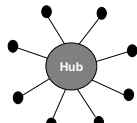
Copyright (A. Umuro) 2002

Integration Approaches (cont.)

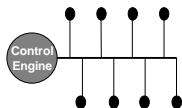
- Tight coupling: use interactive (synchronous), C/S approach
- Loose coupling: use asynchronous approach (e.g., email, messaging)



Point to point
integration



Hub-based
integration

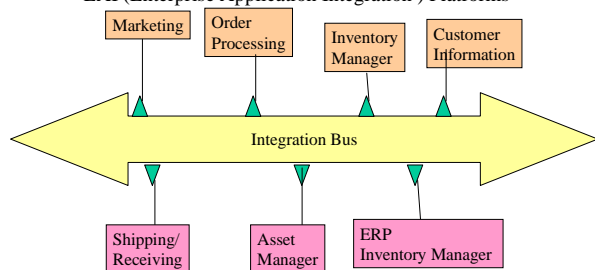


Bus-based integration

- Point to point works fine for small number of applications
- Hub/bus based approaches better for larger apps

Copyright (A. Umar) 2002

EAI (Enterprise Application Integration) Platforms



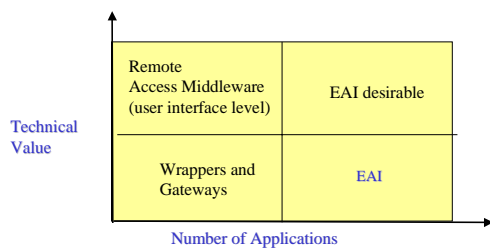
▲ =
Adapter/Data
Mapper

EAI Platforms combine many desirable features:

- Connectivity at data, method, UI level
- Use loosely coupled hub/bus model
- handle message transport differences (e.g., CORBA, MOM, FTP)
- Handle message format differences (EDI, XML, others...)
- Handle diversity of applications (Mainframe, Unix, Windows)

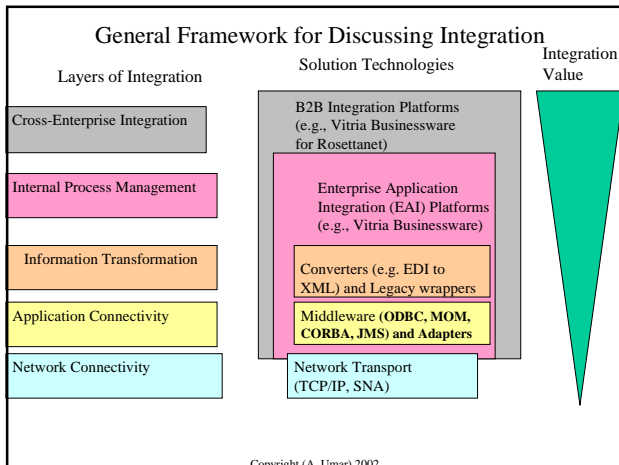
Copyright (A. Umar) 2002

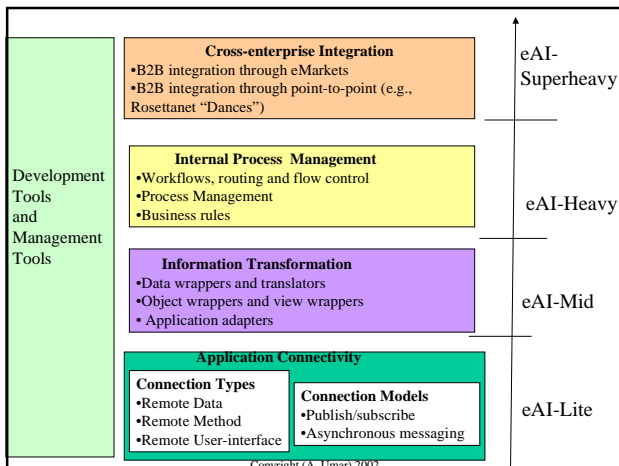
Where to use EAI

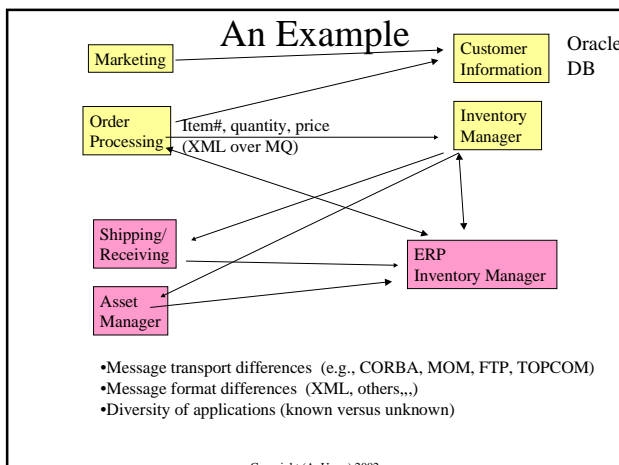


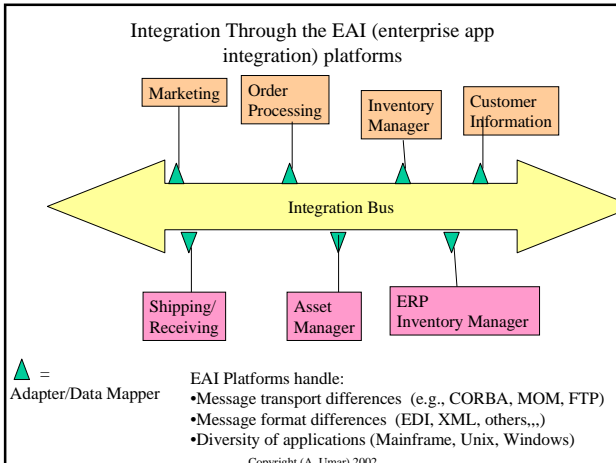
EB is forcing many applications with different technical values (legacies and new web-based) to interact with each other. Thus EAIs are becoming important.

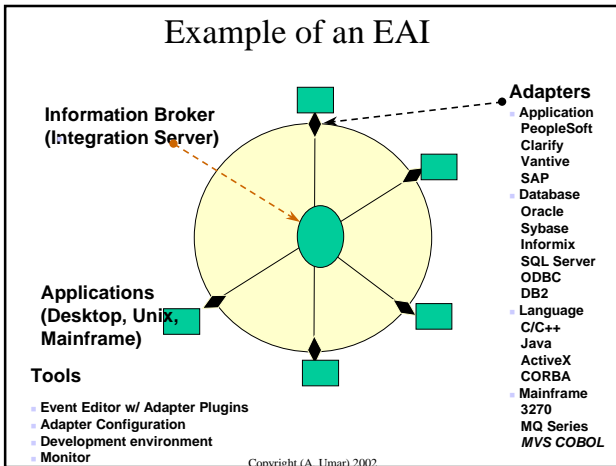
Copyright (A. Umar) 2002

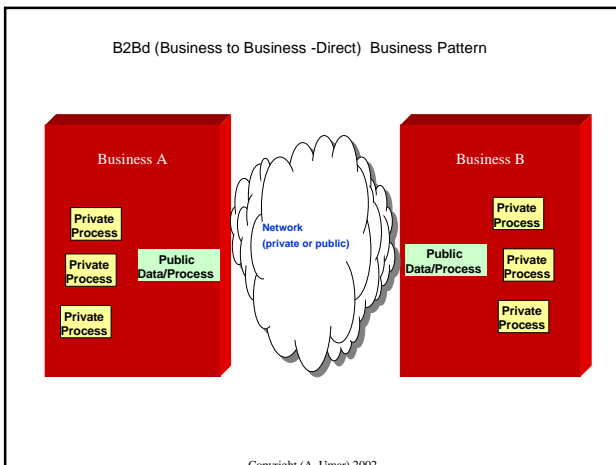


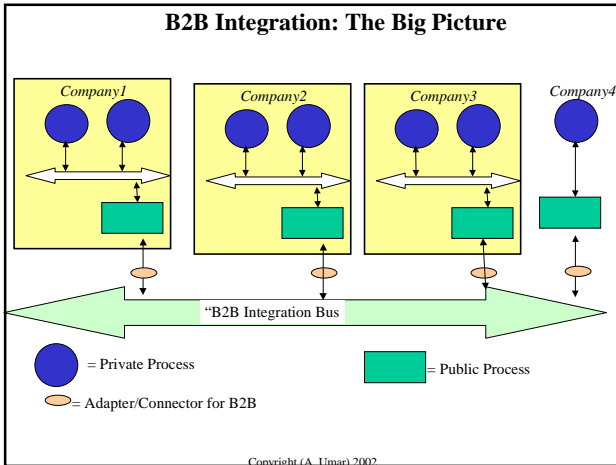










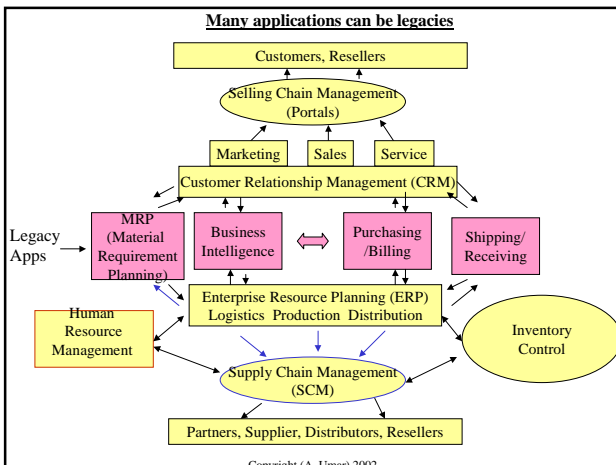


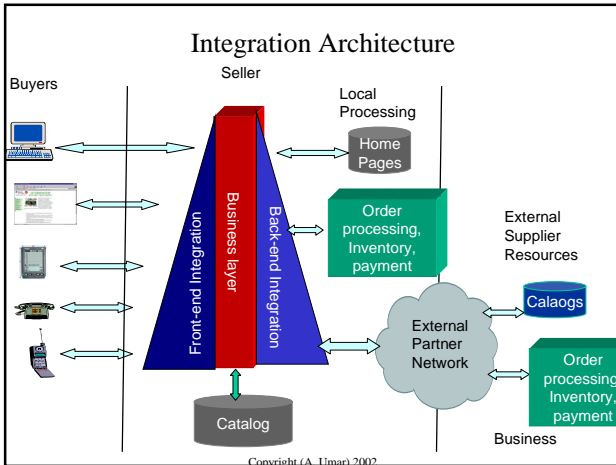
Challenge: Dealing with Legacy Systems

Legacy System Characteristics

- Crucial to the day-to-day operation.
- Large (millions of lines of code)
- Old (5 to 10 years, some older)
- Used heavily (1000s of transactions/day)
- Inflexible, costly, time-consuming.
- Risky to maintain and change.
- Based on older database technology
- Sometimes uses no DB tech at all (VSAM/ISAM files)
- Written in older languages
- Text-based interfaces (3270) not GUI
- Vertically integrated and monolithic
- Reflect old business methods and rules

Copyright (A. Umar) 2002

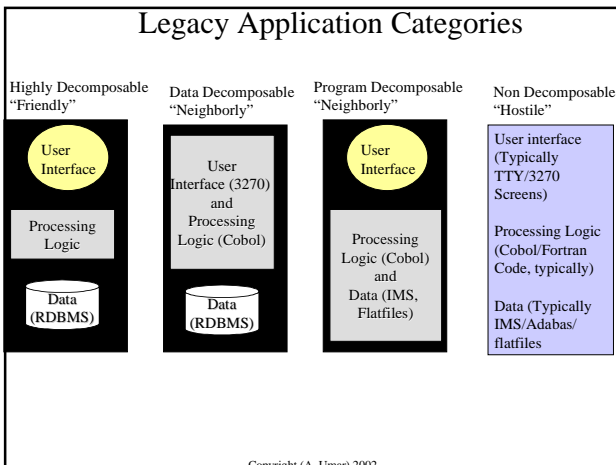




Typical Legacy Systems -- Another Look

	Data Technologies	Program Code	User Interface	TP (Transaction Processing) Monitor
Early 1970s	Flat files (sequential, indexed sequential, VSAM, random)	Assembler and COBOL	Character-based	None or Homegrown
Late 1970s	Non-Relational Databases IMS (Very few use IDMS, Total, and System 2000)	COBOL and PL/1 (Assembler is rare)	3270 Screens	CICS or IMS-DC
Mid to late 1980s	Relational Databases such as DB2 (Very few use Oracle on MVS)	COBOL and PL/1	3270 Screens	CICS or IMS-DC

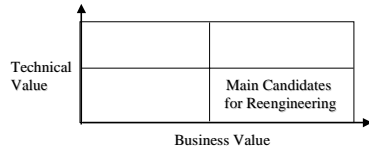
Copyright (A. Umar) 2002



Strategic Analysis

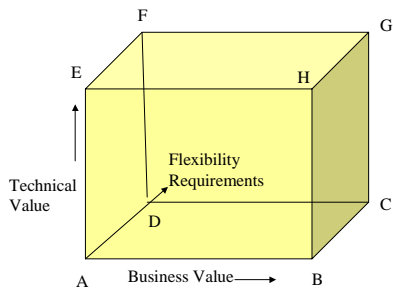
1. Ignore existing (legacy) apps.
2. Rewrite apps from scratch.
3. Provide access/integration through “surround” technologies (mediators).
4. Develop a shadow database (“data warehouse”).
5. Re-architect and migrate.

Real life situations require a mixture of strategies.



Copyright (A. Umar) 2002

Framework to analyze strategies

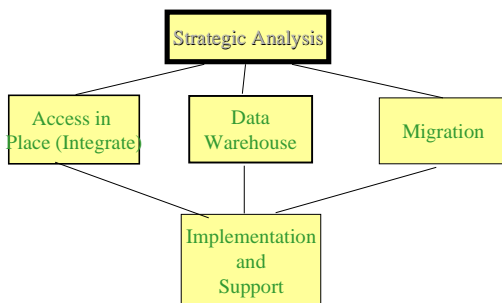


Each point represents a possible strategy

Copyright (A. Umar) 2002

Strategic Analysis

-- Which approach to use (access in place, data warehouse, migration, or mixture)



Copyright (A. Umar) 2002
