

Evolving System Architecture to Meet Changing Business Goals

*An **Agent** and **Goal**-Oriented Approach*



Daniel Gross & Eric Yu
Faculty of Information Studies
University of Toronto
May 2001



1

The Problem

- How to support *evolving* system architectures to meet *changing* business goals.

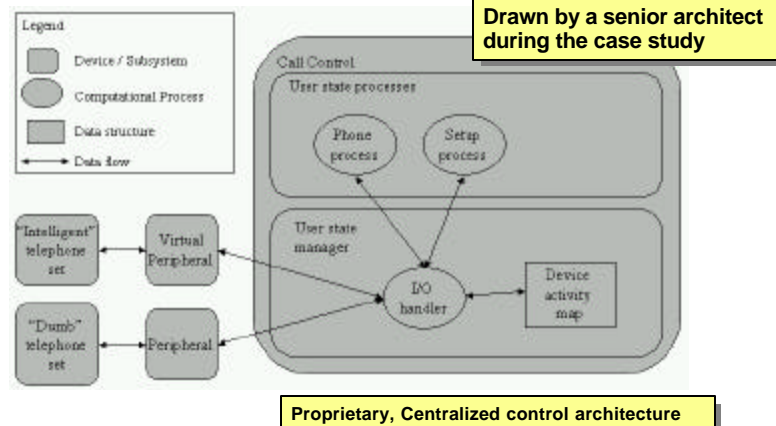
“the bigger picture”

- How to have *goals among agents* drive the design process.

2

Given a Telephone System architecture

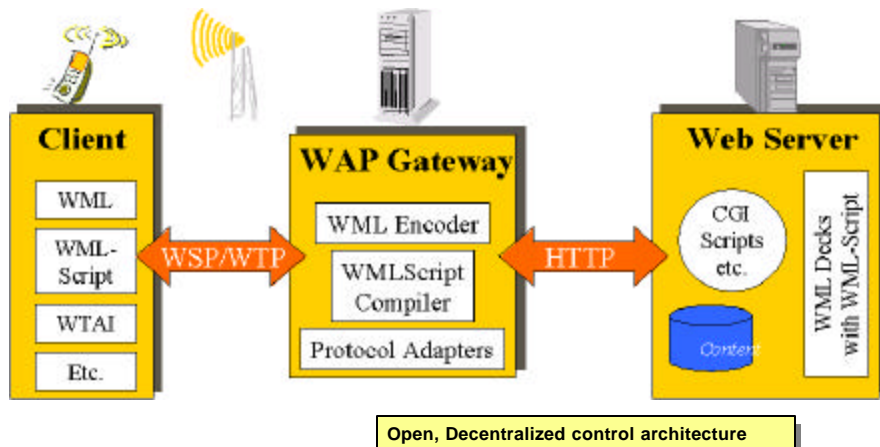
- I know what the system does, however:
 - What business goals led to these architectural structures?
 - What happens to the structures when business goals change?



3

For example: adding internet browsing on the telephone sets through WAP* architecture

It's a business tactic to differentiate the companies telephone set offering through enhancing the ability to design & access internet based service



*WAP – Wireless Application Protocol

4

Where to place the *Client* in the telephone system?

1. Within Call Control ?
(stick to centralized control arch.)

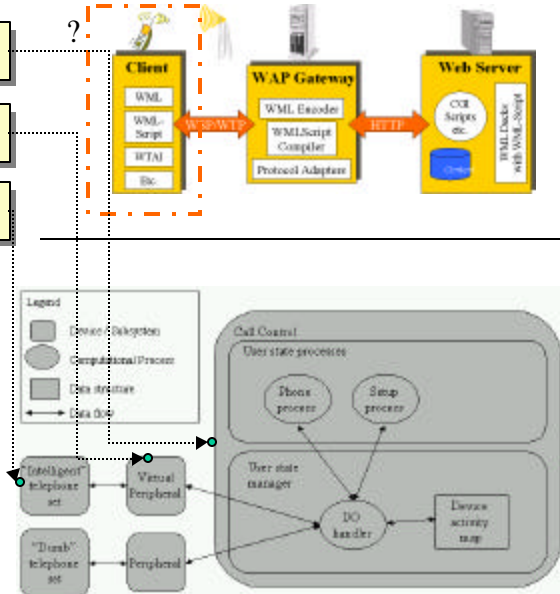
2. Within the Virtual Peripheral?
(towards decentralized contr. arch.)

3. Within the "intelligent" phone
set? (decentralized control arch.)

More generally:
Where to place other
future applications in
the telephone system ?

How to make a decision
without goals?

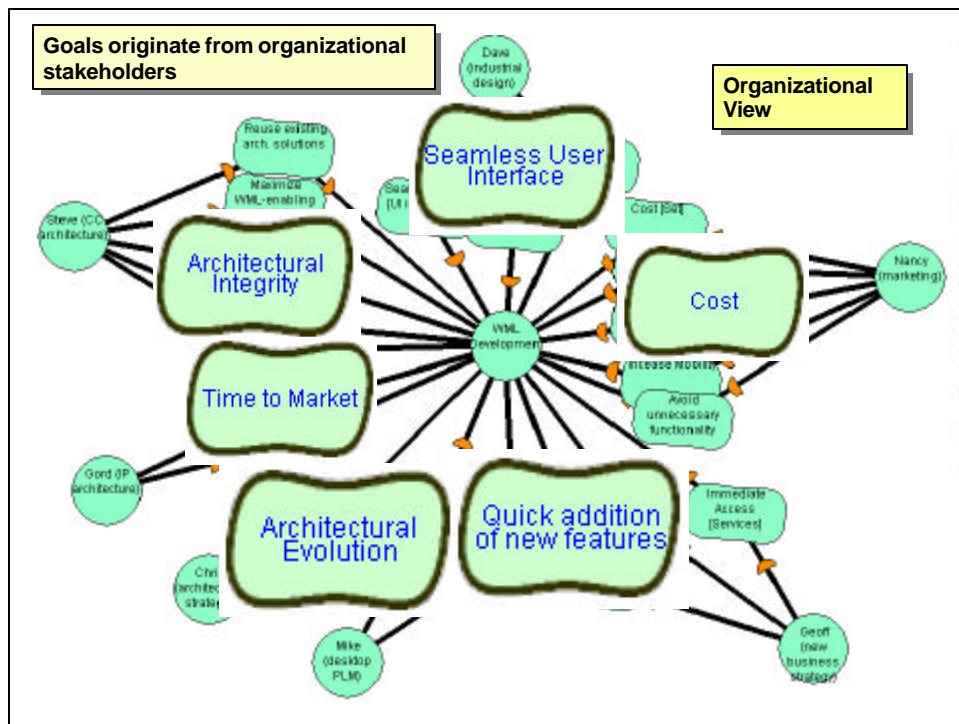
Who cares about the
alternatives and why?



Goals originate from organizational
stakeholders

Organizational
View





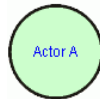
Modeling assumptions

How to model architecture *during* design ...

- when requirements notations drive architectural notations [Mylopoulos, STRAW]
- when acknowledging that
 - architecture of a system is a “living” dynamic evolving “organism”
 - the *design process* never ends but “spirals” up and down
 - architecture design & evolution is a *social* negotiation process

Actor Notation

Actors = (capabilities+ Goals) that *eventually* become components or connectors in the “finished” design



Actor A denotes some design unit under development.



For example:

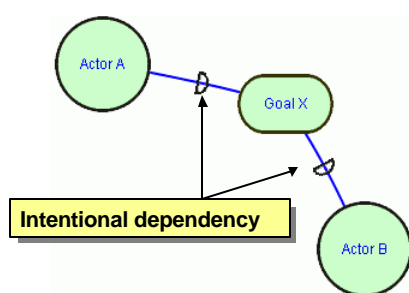
Denotes the “*new application*”, such as the WAP client, to be introduced into the current architecture

We wish to show how goals are propagated among actors *during* design !

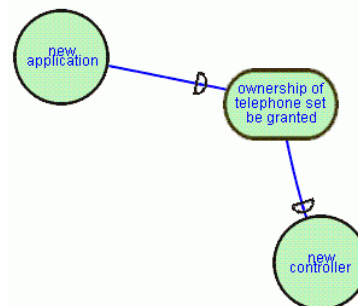
9

Intentional Goal Dependency

Actors = (capabilities+ Goals) that *eventually* become components or connectors in the “finished” design



Actor A depends on Actor B to achieve Goal X during further design.

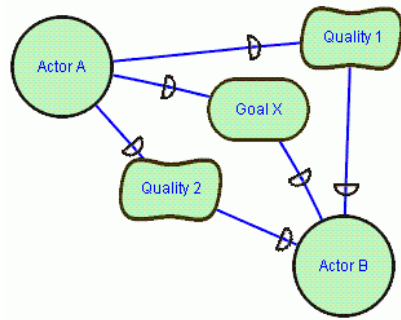


“new application” expects the “new controller” to be designed such that it can grant ownership to a shared telephone set (not shown).

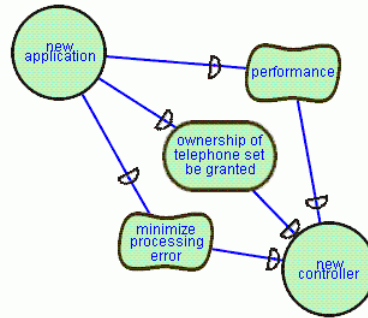
10

Intentional Softgoal Dependency

Actors = (capabilities+ Goals) that *eventually* become components or connectors in the “finished” design



Actor A depends on Actor B to achieve Qualities 1,2 while achieving Goal X.

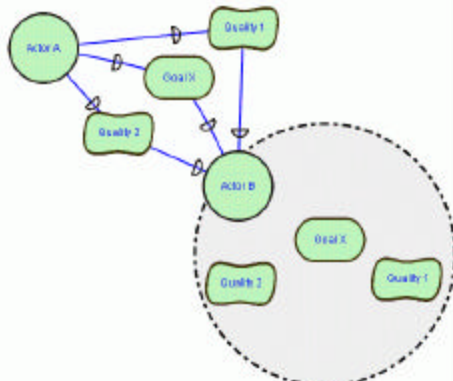


“new application” expects the “new controller” to be designed such that its performance is not degraded and that no processing errors occur during controlling.

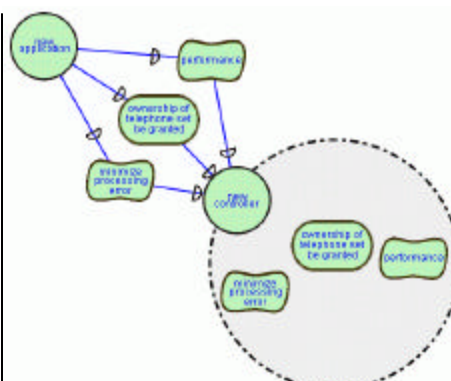
11

Actor Internal View

Actors = (capabilities+ Goals) that *eventually* become components or connectors in the “finished” design



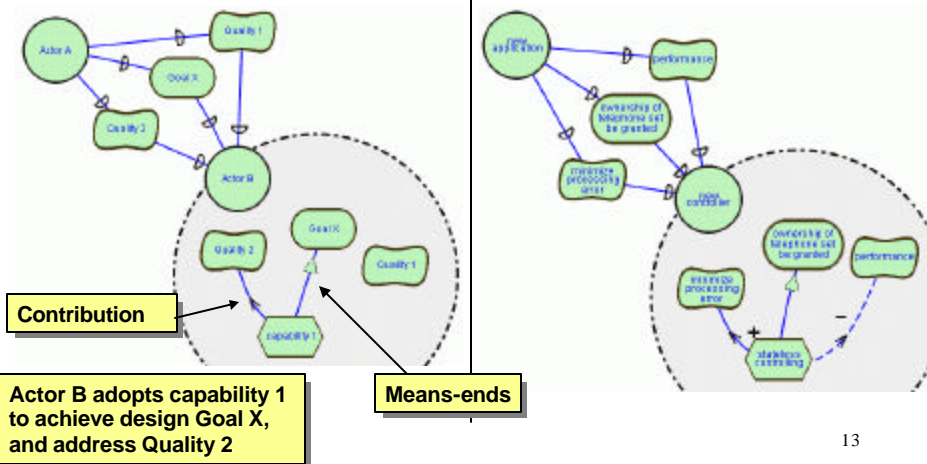
Actor B needs to achieve design Goal X, Qualities 1, 2 by designing some capabilities.



12

Capabilities and Goals

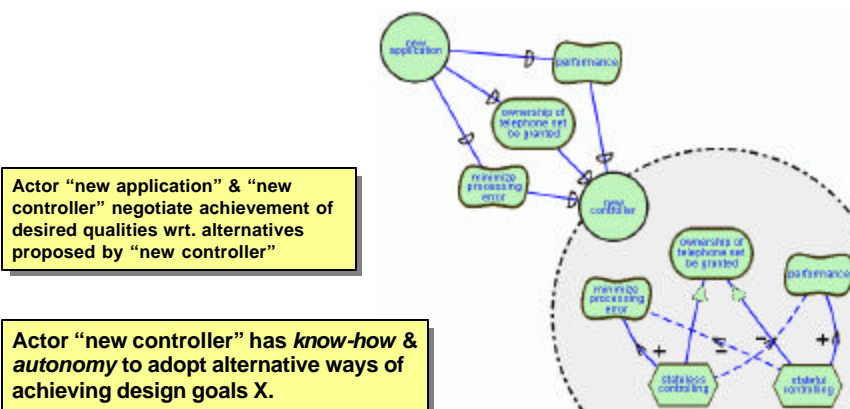
Actors = (capabilities+ Goals) that *eventually* become components or connectors in the “finished” design



13

Alternatives during design

Actors = (capabilities+ Goals) that *eventually* become components or connectors in the “finished” design



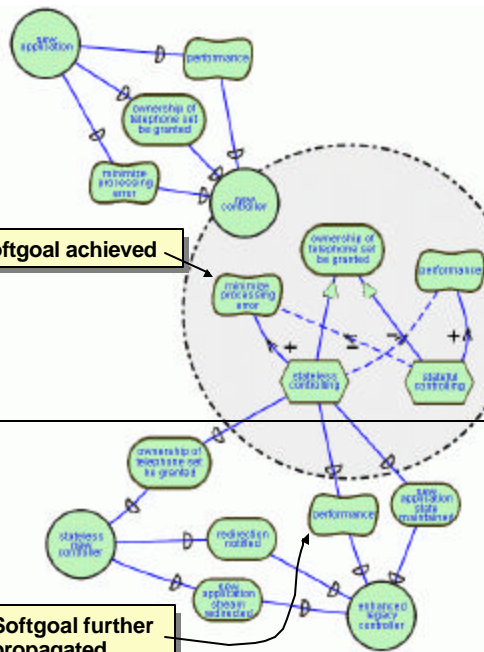
14

Actors establishing new Actors

Distribution of design goals
based on the stateless
controlling alternative

Softgoal achieved

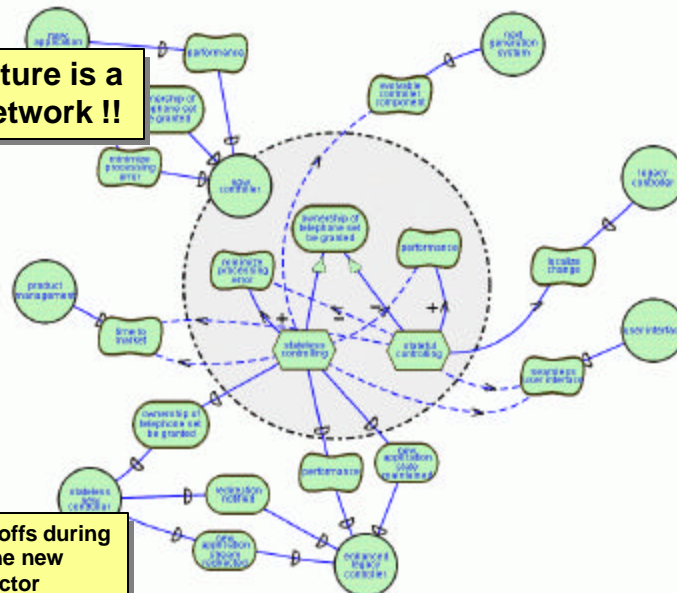
Softgoal further
propagated



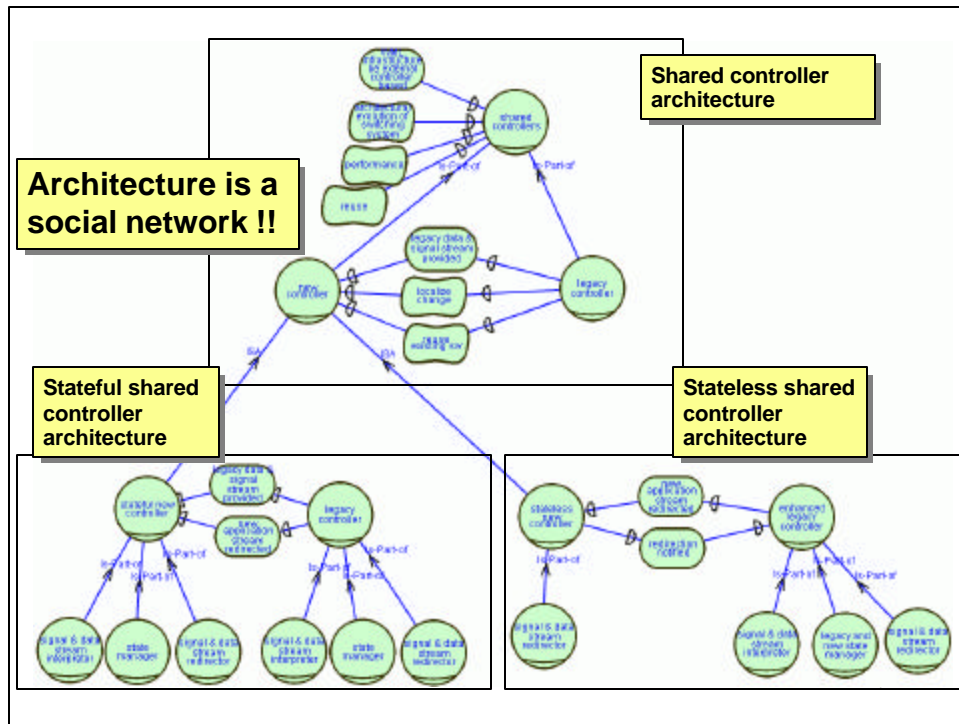
Additional intentional dependencies

Architecture is a
social network !!

Some tradeoffs during
design of the new
controller actor



16



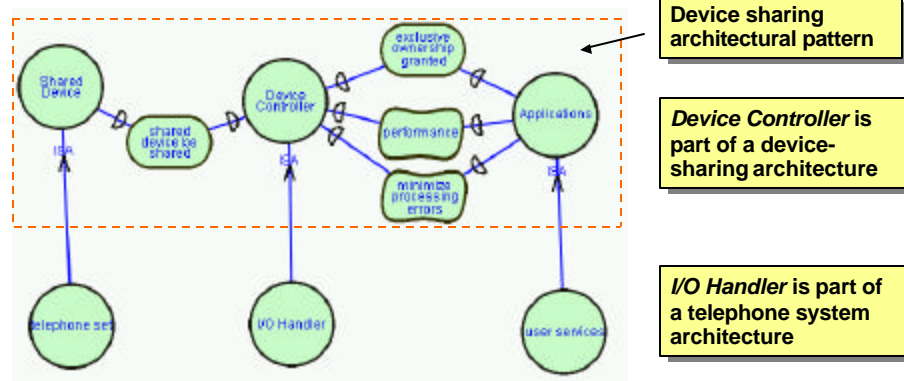
Conclusions & Future work

- Treating architectural elements as Actors allows
 - Introducing, distributing, negotiating and tracing goals and their achievement by architectural elements during the design process and during evolution.
 - Provides the basis for goal driven design guidance
- Better integration of modeling views needed
- Methodological support
 - Also possible integration into Boehm et. al. work related to negotiations
- Stakeholder oriented viewpoints
 - Management view, designers view, etc.
- Actor/Agent extension for ITU-URN/GRL effort

Supplements

19

Reusing architectural fragments through “ISA” links



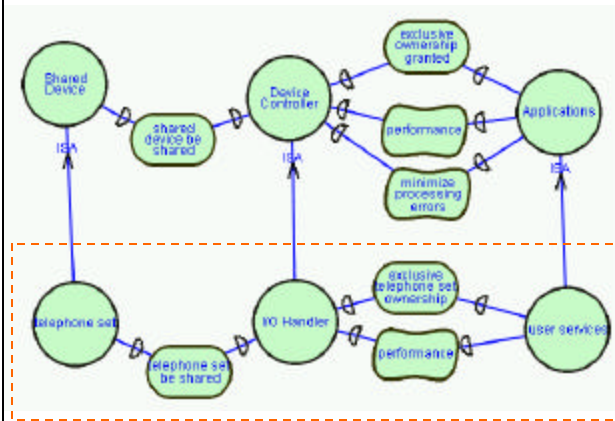
The designer of the *I/O Handler* might now:

- Grant ownership to user services
- Deal with *Performance* and/or *Minimizing processing errors* to keep the *user services* actor happy.

Note: Creating ISA links is a step in the design process

20

Intentional dependencies are inherited



Note: Inheriting intentional dependencies is a design step ...

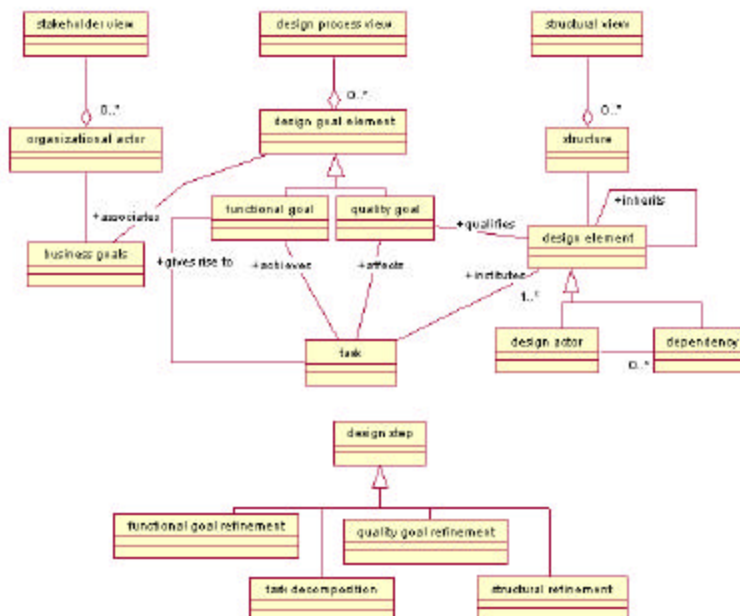
... done *interactively* and *selectively* together with *rationales*

which are recorded in the process view

**Telephone system
architecture fragment**

21

Modeling Views relationships



22

