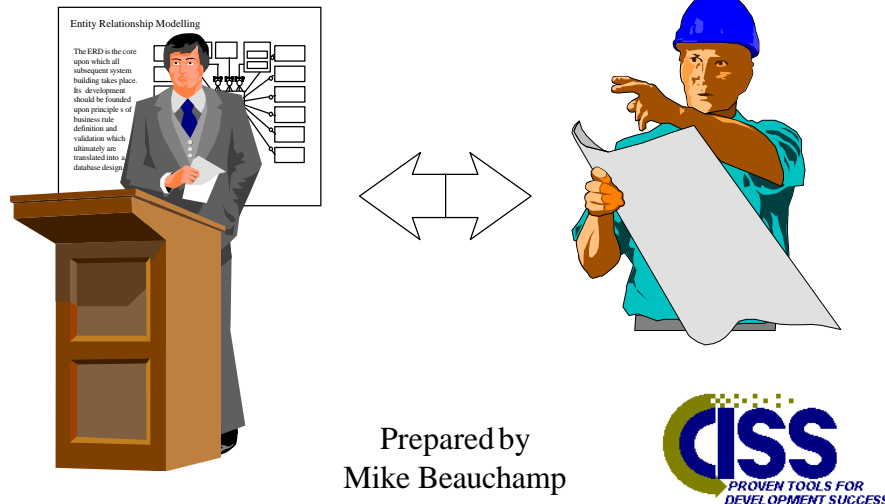


**Track VI: DESIGN**  
**Session 600**  
**Data Modeling for**  
**Buildable Components**



## Regulatory 95 Project

**Re-architect existing Regulatory Systems (6 discrete systems).**

**Major Objectives:**

- improve the efficiency of regulatory investigation process
- improve timing accuracy of trade activity in accordance with Futures Trade Practices Act of 1992

**Target technology:**

- OS/2 clients with multiple Servers (MVS/DB2, RS6000/Oracle)

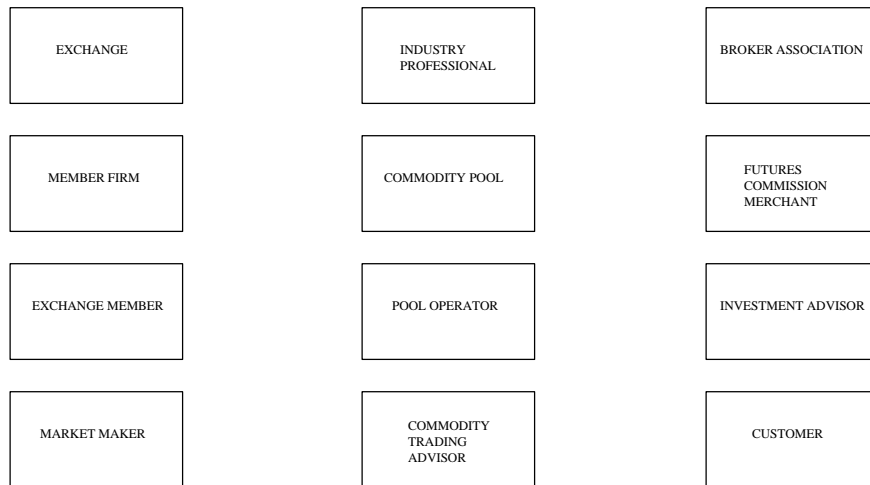
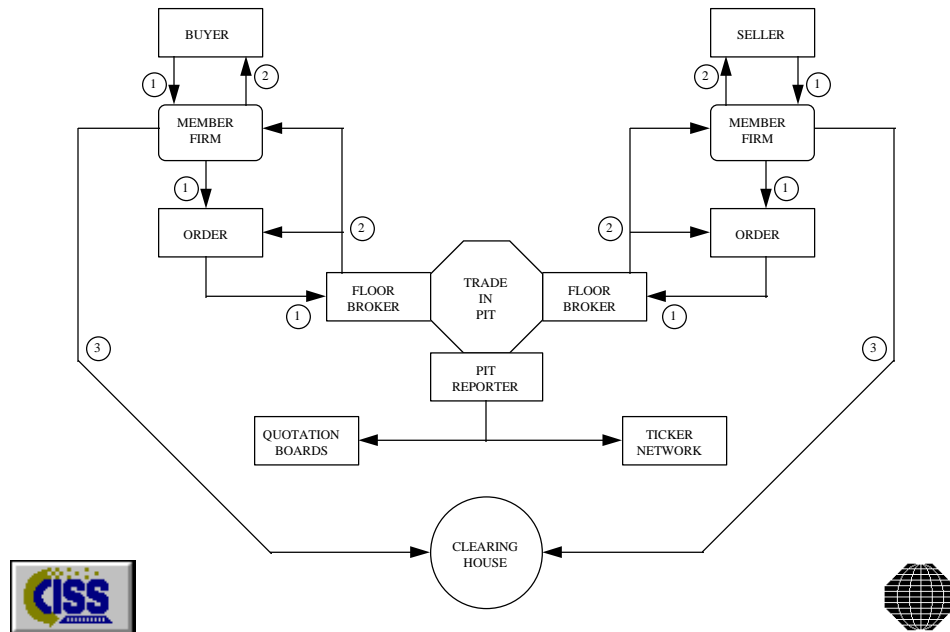
**Use IEF to define integrated database and develop core applications.**

**Data Model has 20 business Subject Areas and approximately 170 entities.**

Chicago Mercantile Exchange



## Trading Dynamics

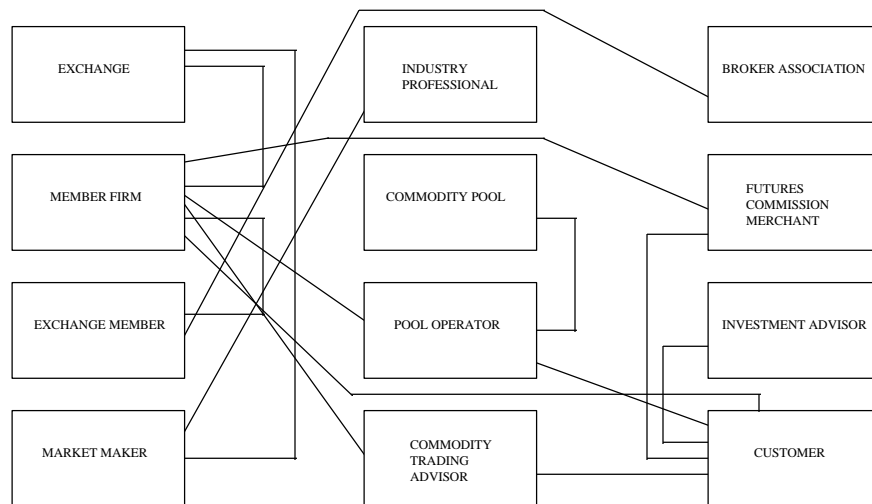


## Business Entity Definition



Defines what is of specific concern; the 'real' things.

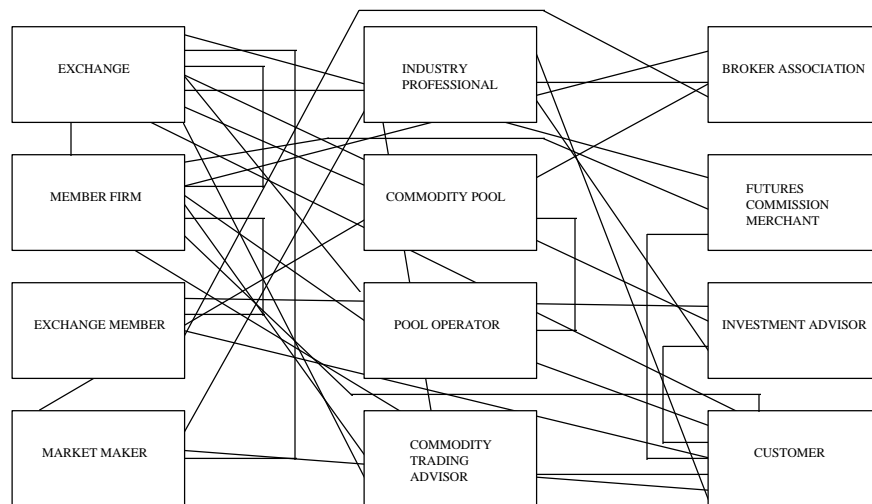




### Business Relationship Definition



‘Real’ relationships can get **real** complex.

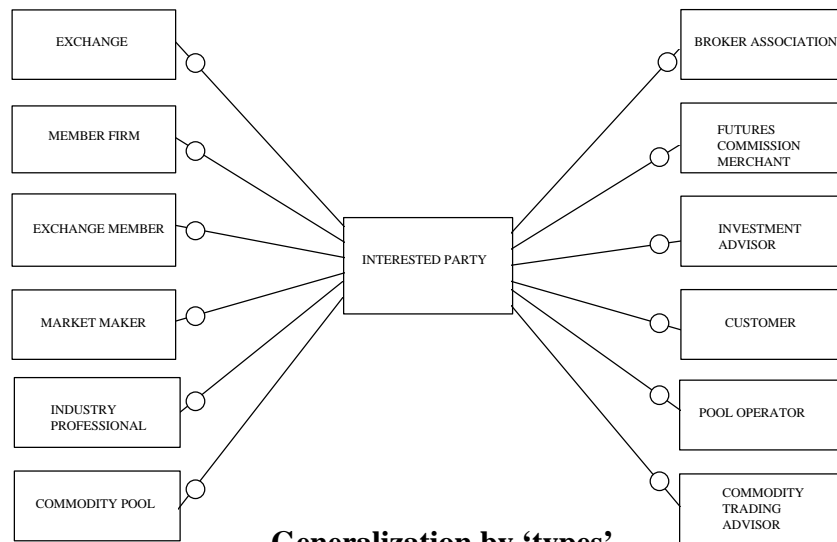


### Business Model Evolution



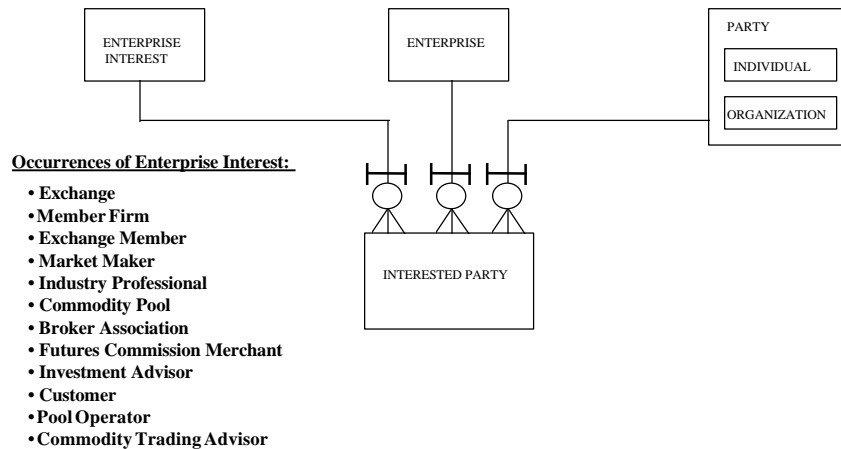
But can we build anything with it?





### Generalization by 'types'

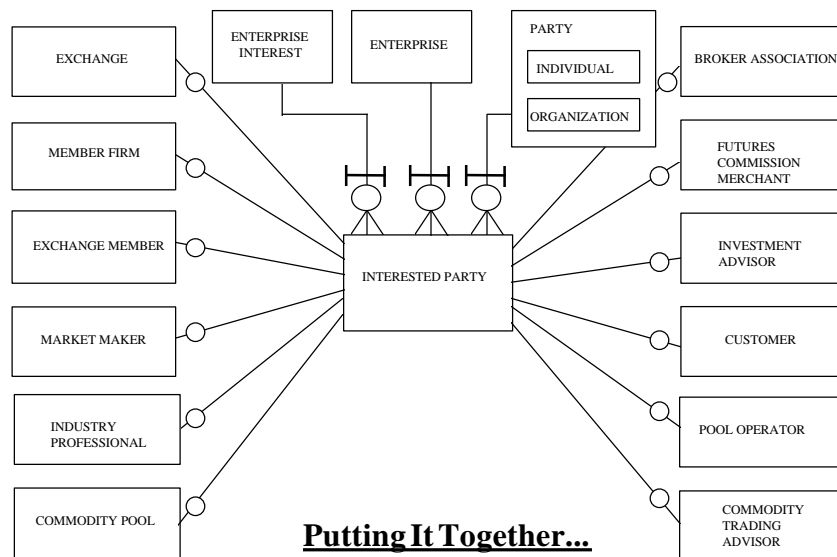
Introduction of the 'Interested Party' concept.  
But is this enough?



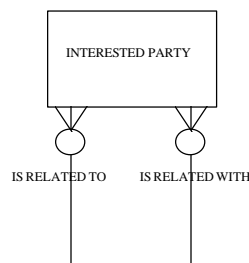
### Normalize the Generalization

A *Party*, being either an *Individual* or an *Organization* becomes an *Interested Party* when they possess an *Interest* in an *Enterprise*.





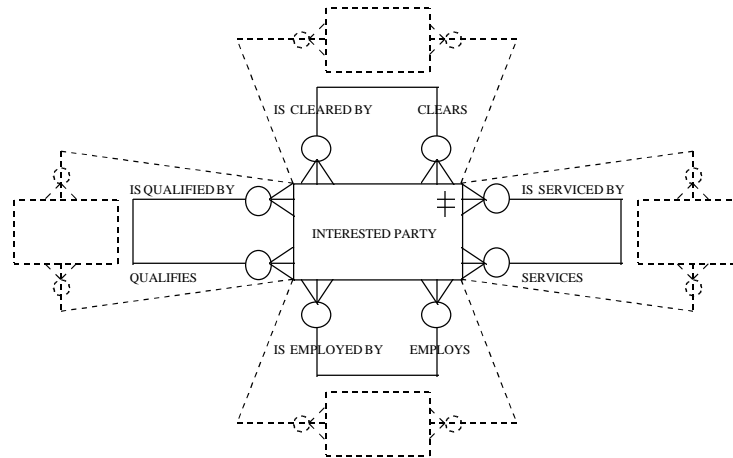
A Party's Interest in an Enterprise is performed through an Interested Party's '**participation**' as a specific, 'real world' entity.



### **Related Interested Parties**

This is a 'generalized' relationship. The relationship labels give no business context. So what happens when we use 'real' relationships?....





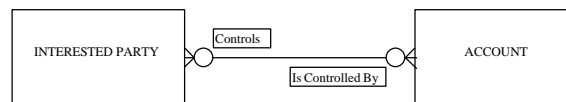
### **Resolving Multiple Many-to-Many Relationships**

Just how many associative entities and resulting tables and indices are you willing to tolerate?  
There is another way.



### **Many-to-Many Relationships**

Here's an example....

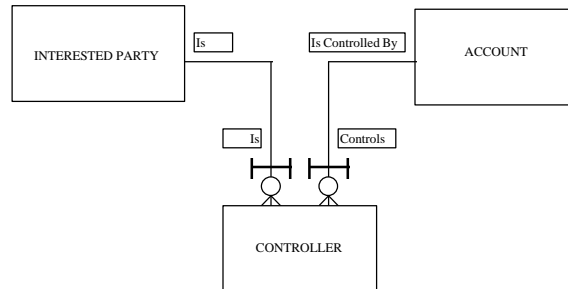


We resolve this by introducing  
an associative entity....



## Associative Entity Resolution

The Account Controller.

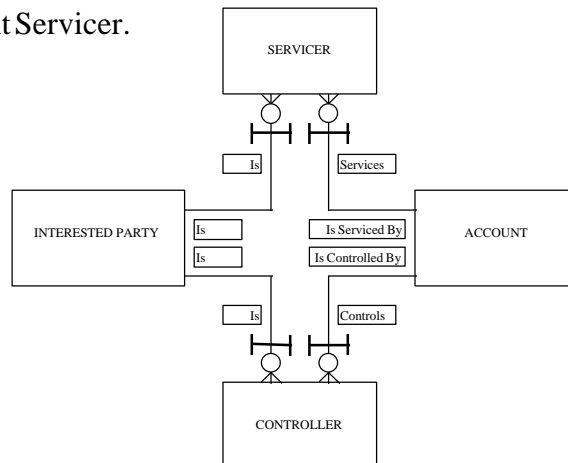


There are also Interested Parties who  
'Service' accounts, leading to....



## Multiple Resolutions

The Account Servicer.

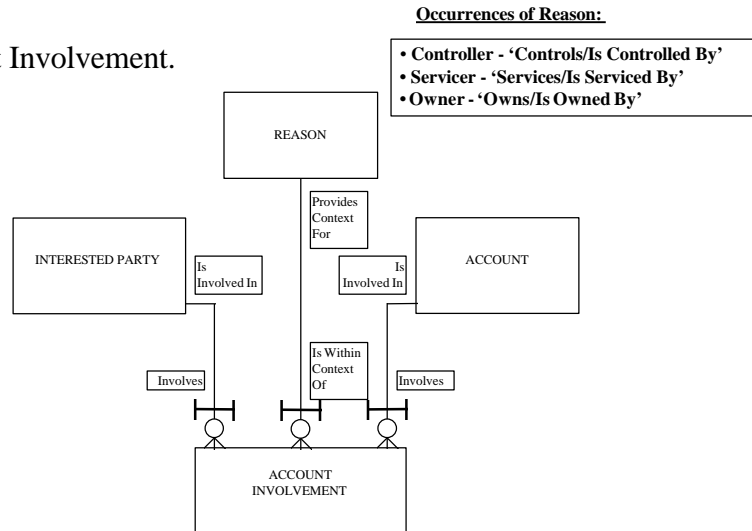


There are many similar resolutions possible,  
but all of them can be generalized to....



## Generalized Resolution

An Account Involvement.

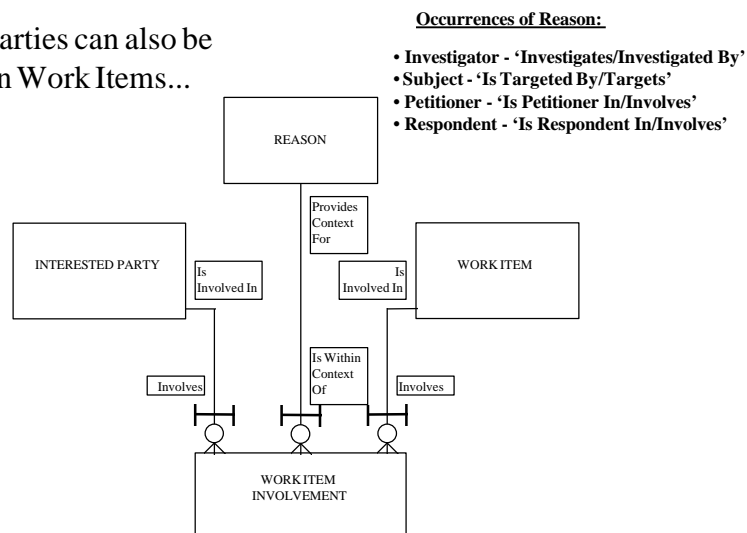


The REASON entity provides the business context being managed by an occurrence of INVOLVEMENT.



## Consistent Application

Interested Parties can also be 'involved' in Work Items...



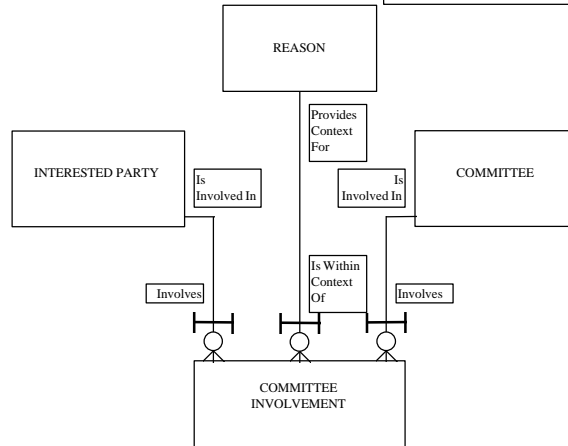


## Consistent Application

Committees...

### Occurrences of Reason:

- Chairman - 'Chairs/Is Chaired By'
- Member - 'Serves On/Is Comprised Of'

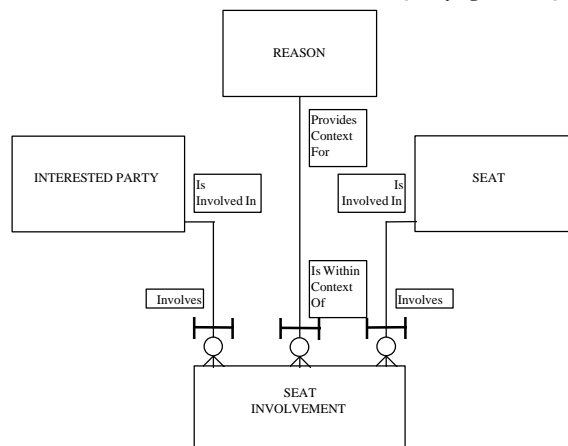


## Consistent Application

Seats...

### Occurrences of Reason:

- Owner - 'Owns/Is Owned By'
- Holder - 'Holds/Is Held By'
- Qualifying Firm - 'Qualifies/Is Qualified By'



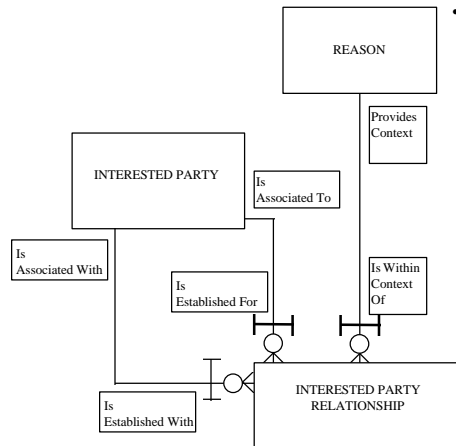
It can be applied between any two principal entities requiring associative entity resolution.



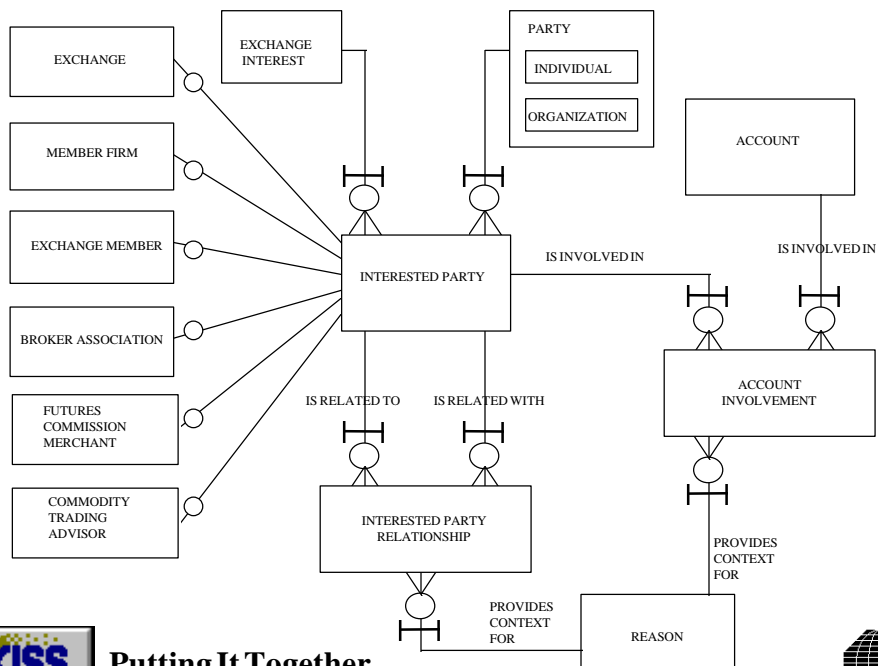
## Consistent Application

### Occurrences of Reason:

- 'Employs/Is Employed By'
- 'Qualifies/Is Qualified By'
- 'Clears/Is Cleared By'



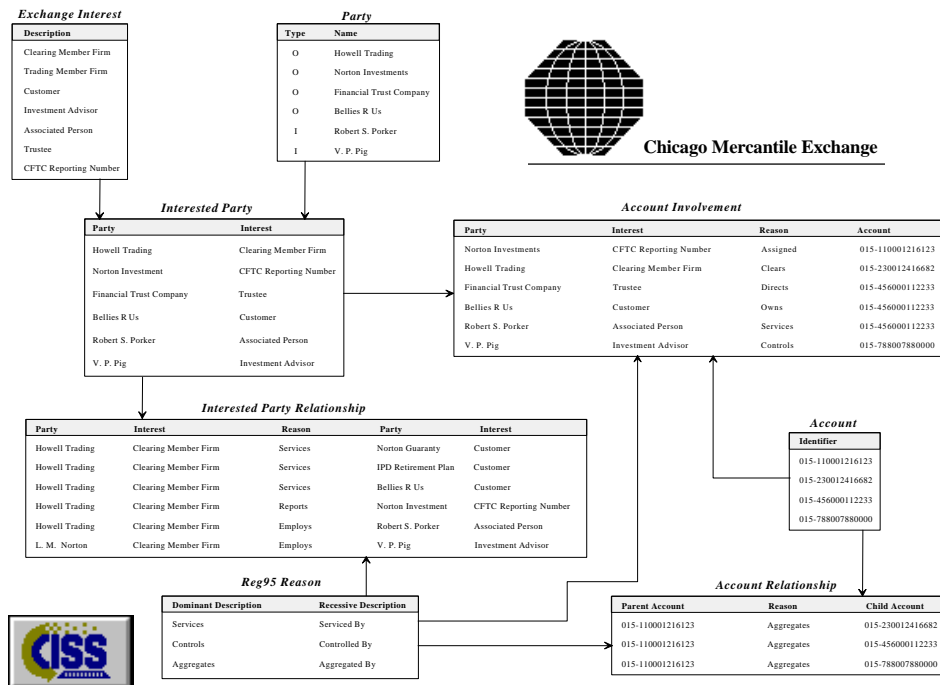
It can also be used to resolve involuted relationships.



Putting It Together...



## Interested Party and Accounts Data Occurrence Diagram

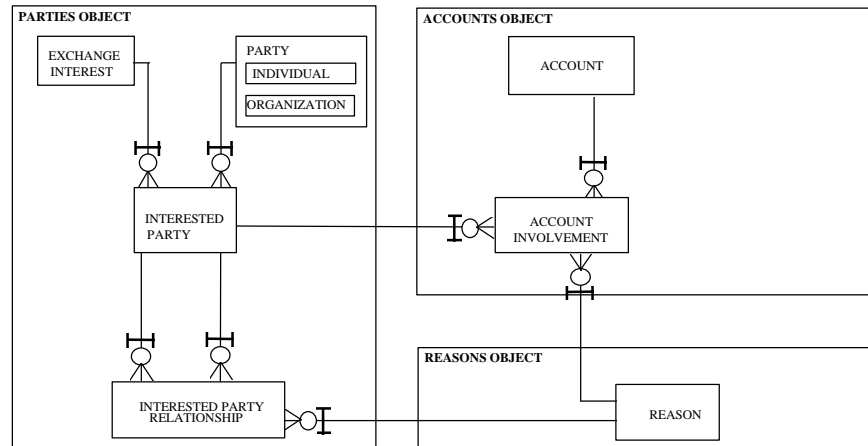


## Impact of Approach

- ☆ 10 'Involvement' type entities defined using approximately 40-50 known occurrences of *Reason*.
- ☆ Requires 30 less tables and 50 less index structures in the resulting database when compared to the use of specific resolution.
- ☆ System enhancements to support additional 'involvements' or 'relationships' do not require database changes.
- ☆ Effective use of *Interested Party* eliminates multiple foreign keys for mutually exclusive relationship sets.
- ☆ Consistency of approach supports reusability of developed components.
- ☆ Understanding the business significance of *Exchange Interest* and *Reason* will yield powerful querying capabilities.
- ☆ You may need to maintain other, more specific 'logical/conceptual' versions of the model to support a wider audience for strategic acceptance and understanding.



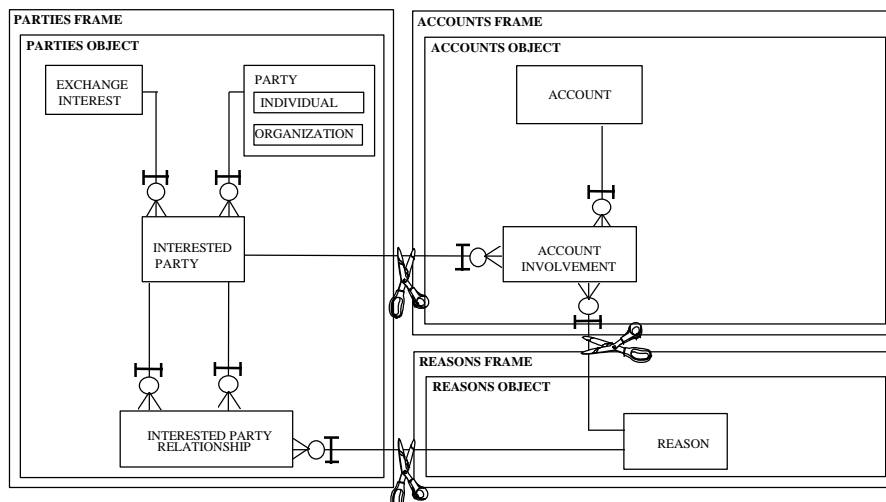
## Adjusting the Model for Components



Organize subject areas so that they implement “one” central business entity and its dependents.



## Adjusting the Model for Components



Define independent *frames* and remove relationships that cross frame boundaries. Add ‘foreign keys’ as attributes instead.



## Removing Relationships - Points to Ponder

Impact to Referential Integrity

Re-thinking Referential Integrity

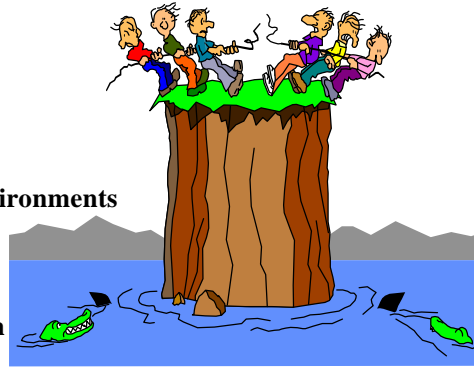
Complexity in Action Diagramming

Need to support separate server environments

DBMS vs. IEF enforced RI

Multiple vs. Single Technical Design

IEF tool shortcomings (yes, there are some)



Ask yourself one question. Do I really *need* to support development of independent frames?

