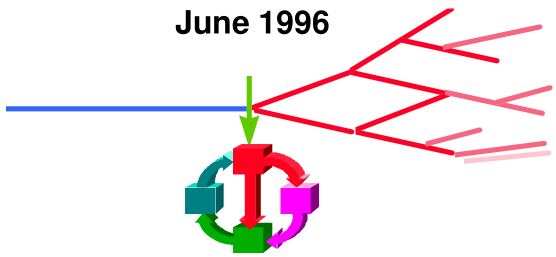
draft

# SimQL: Accessing Simulation as Services

Gio Wiederhold

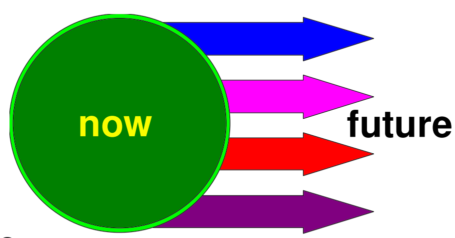
June 1996





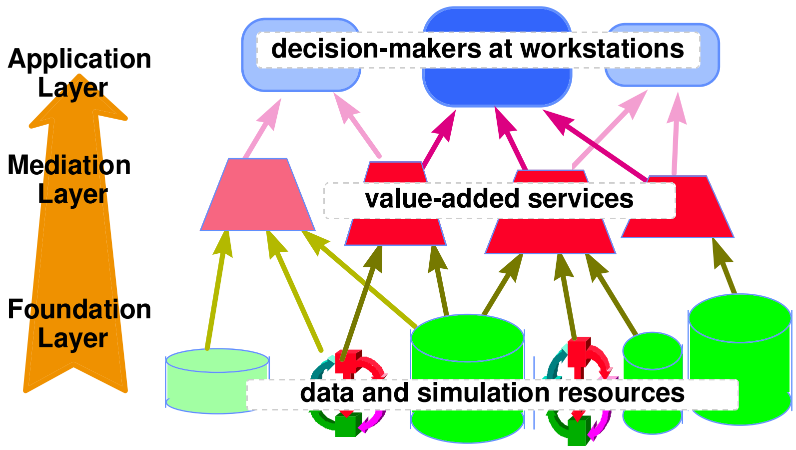
## Decision-making (DM)

**Analyze Alternatives**
- Current Capabilities
- Future Expectations



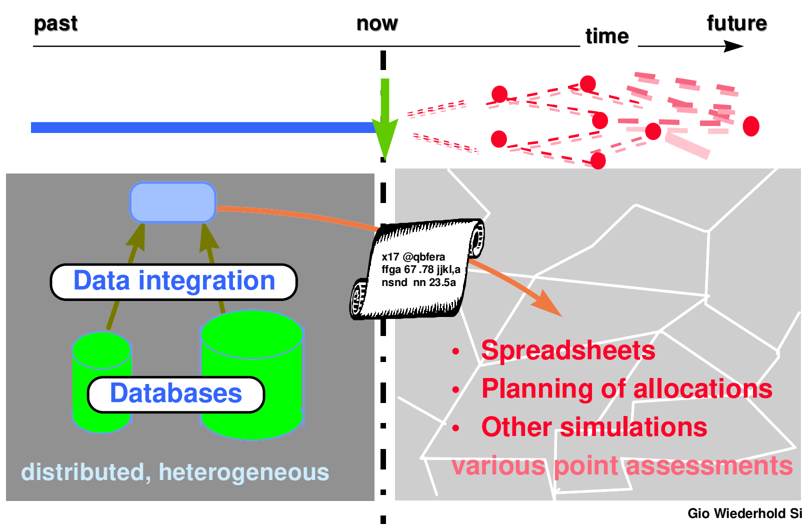

**Process tasks:**
- List resources
- Enumerate alternatives
- Prune alternative
- Compare alternatives

# Transform Data to Information



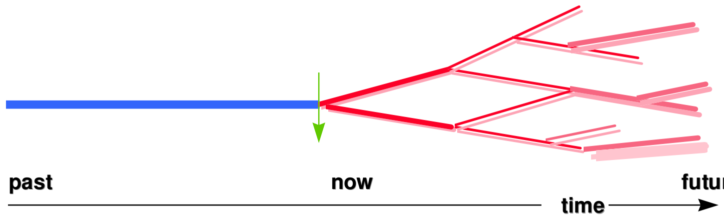

Gio Wiederhold SimQL 3

# Current state of DM Support




Gio Wiederhold SimQL 4

## Information Systems should also Project into the Future



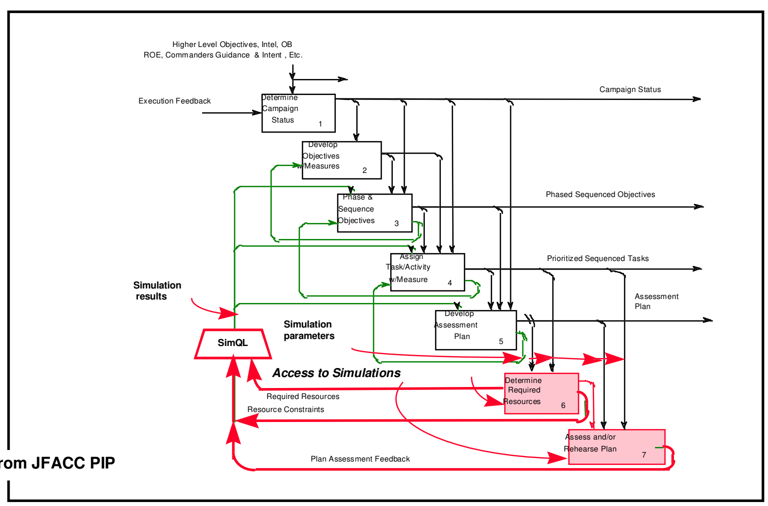

**Support of decision-making requires dealing with the future , as well the past**

- **Databases deal well with the past**
- **Sensors can provide current status**
- **Spreadsheets, simulations deal with the likely futures**

**Information systems should be able to combine all three**

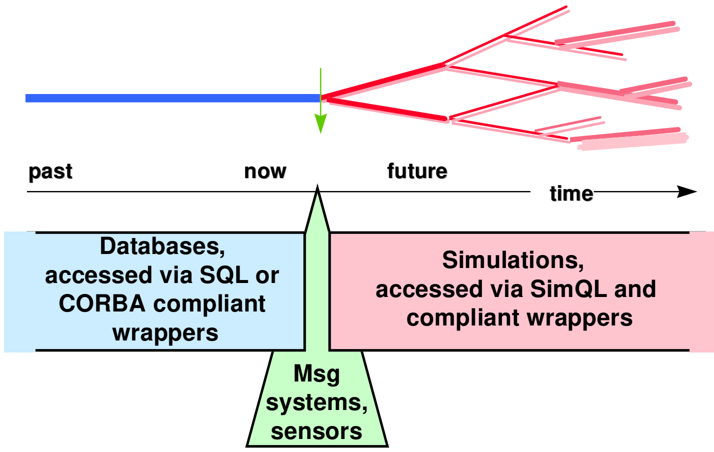


## Place of SimQL in Objective-based Planning




**from JFACC PIP**

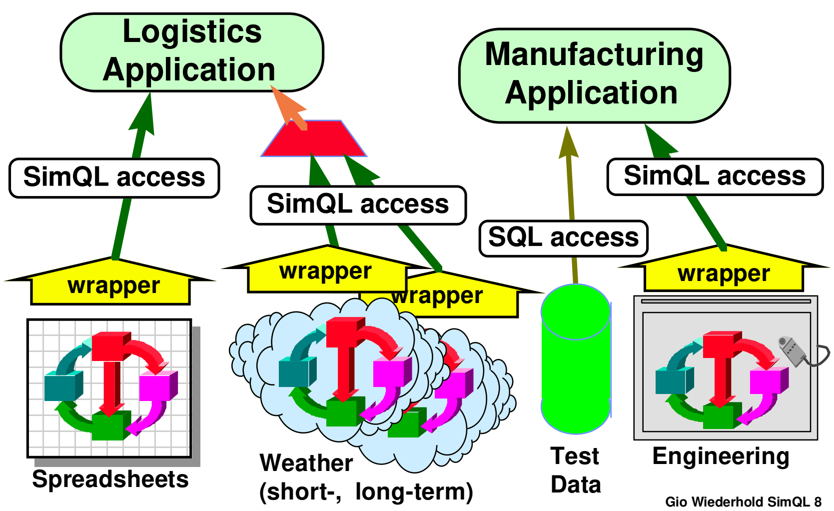
# Information Systems should also Project into the Future




Gio Wiederhold SimQL 7

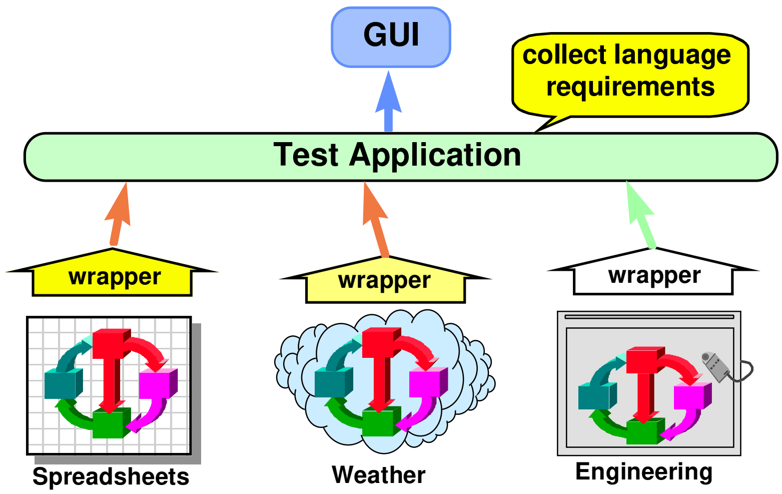
Stanford experiment, supported by DARPA & NIST

# Phase 1 Architectures




Gio Wiederhold SimQL 8

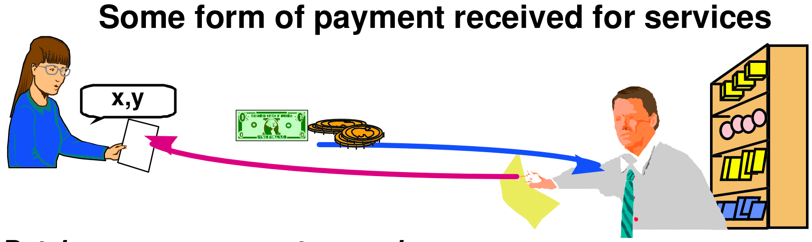
## Current State of SimQL research






## Moving to a Service Paradigm

- Server is an independent contractor, defines service
- Client selects service, and specifies parameters
- Server's success depends on value provided
- Some form of payment received for services




*Databases are a current example.*
*Simulations have the same potential.*

## Enabling Interoperation

**Databases**

- **serve clients via SQL by**
  - Sharing a Model (The Schema)
  - A query language over the model

**the SQL interface enables**

- **independence of**
  - application development
  - DBMS technology development
  - reuse of infrastructure

**Today**

- **most new systems use a DBMS for data storage**
  - even with less performance,
  - inability to handle *all* problems,
  - but enough of them well enough.

**Simulations *should***

- **serve clients via *SimQL* by**
  - Sharing a Model (*research* q.)
  - A query language over the model

**a *SimQL interface* will enable**

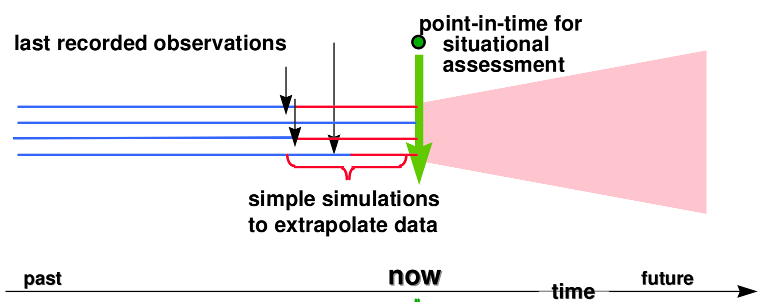
- **independence of**
  - application development
  - simulation technology develop't
  - reuse of infrastructure

**Objective**

- **build information systems combining DBMS, Simulations**
  - even with less performance,
  - inability to handle *all* problems,
  - but enough of them . . .

Gio Wiederhold SimQL 11

## Even the present needs SimQL

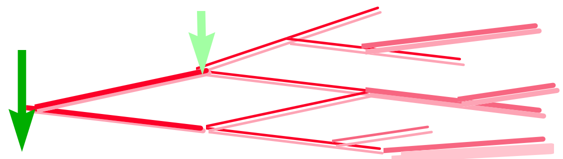



Not all data are current::

Is the delivery truck in X?
• Is the right stuff on the truck?
• Will the crew be at X?
• Will the forces be ready to accept delivery?

Gio Wiederhold SimQL 12

## Use of Simulation Results



Simulation results can be composed for Alternative Courses-of-actions

Composition should be seamless, elegant, with computation and recomputation of likelihoods

Results change as now moves forwards and eliminates earlier alternatives.



## Types of simulation services

1. **Continously executing**: weather prediction
   – SimQL result reports best match samples
2. **Execution specific to query**: what-if assessment
   – may require HPC power for adequate response
3. **Past simulations collect results in a base**: materials
   – performs inter- or extra-polations to match query parameters
4. **Combinations**, i.e., 2. + 3.: top layer simulation using stored partial lower level results: weapon performance in setting
5. **Human-in-the-loop** (mediated by an agent program): SAFs

Note

- A simulation service program can be written in any language
- A simulation service must be compliant to the interface spec.

## Paying for Simulation Services

You can not run an effective (SW) business and not be reimbursed for it. How? Four approaches:

| | |
|---|---|
| • Buy Software | *buy oilfield* |
| • Lease copy / usage rights | *lease well* |
| • Time / user limited access | *fill tank* |
| • Pay by use instance | *take public transit* |

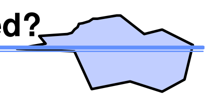
General problems, differ by approach

| | |
|---|---|
| – copy protection? | *B - -, L -, T+, P++* |
| – keeping SW updated | *B - -, L -, T++, P++* |
| – accounting and billing | *B++, L+, T -, P - -* |
| – performance | *B++, L+, T -, P - -* |



## SimQL research questions

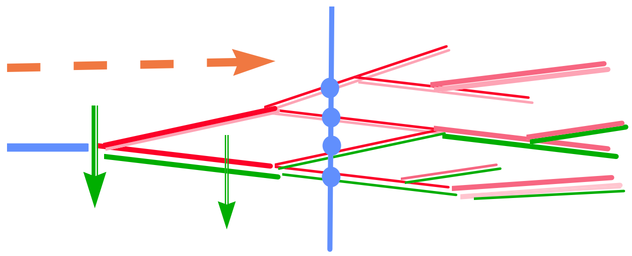
- How little of the model needs to be exposed?



- How can defaults be set rationally?
- How should expected execution cost be reported?
- How should uncertainty be reported?
- Are there differences among application areas that require different language structures?
- Are there differences among application areas that require different language features?
- How will the language interface support effective partitioning and distribution?

## New DM research questions

- How to move seamlessly from the past to the future?
- How can multiple futures be managed (indexed)?
- How can multiple futures be compared, selected?
- How should joint uncertainty be computed?
- How can the *NOW* point be moved automatically?



Gio Wiederhold SimQL 17

## Initial JFACC experiment proposed: Situational projection in TacOps

Objective: Rapid summary of alternative CoAs by J5

- Focus on resource (personnel, materiel) usage
- Short term projection: present, +3h, 6h, +1day

Criteria: Ease of use, no extra load

- Graphic output in standard PC formats
- Input derived from messages, orders, ..., by J3,J4

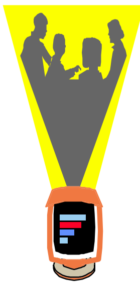
Simplicity in visualization:

- Projected resources (aggregated) on area maps
- Intel-derived (photo-recon) overlays, selected by J2

Benefit: Consistent vision, important in Joint Ops

Gio Wiederhold SimQL 18

# Summary of SimQL



**A new service for DM:**

- **follows database paradigm**
  - (by about 25 years)
- **coherence in prediction**
  - displacement of ad-hoc practices
- **seamless integration over time**
  - single paradigm for decision makers
- **systematic partitioning**
  - better software management
- **simulation industry infrastructure**
  - investment has a potential market