A General and Unified Design and Framework for Distributed Optimization

Robert Fourer Jun Ma

12/12/2003

Industrial Engineering and Management Sciences

NORTHWESTERN
UNIVERSITY

Outline

- Abstract
- Background
  - A General Picture – The Future of Computing
  - Our Positioning – The Hierarchy of Operations Research
  - Some Terminologies (Parallel/Distributed/Grid Computing, XML, Web Services, etc.)
- Motivation
  - Motorola's VP Intelligent Optimization System
  - AMPL-NEOS
- The Design and Framework for Distributed Optimization (NSF Funded)
  - Proposing and Defining Optimization Services (OS, OSXL)
  - Design of Distributed Optimization Architecture (Centralized and Decentralized)
  - Optimization Services Representation (OSTL, OSRL, OSOL, OSSL, OSAL)
  - Optimization Services Communication (OSCL, OSDL, OSFL, OSEL)
  - Optimization Services Inspection and Discovery (OSIL, OSPL, OSBL, OSQL)
- Future Work and Conclusion

Abstract

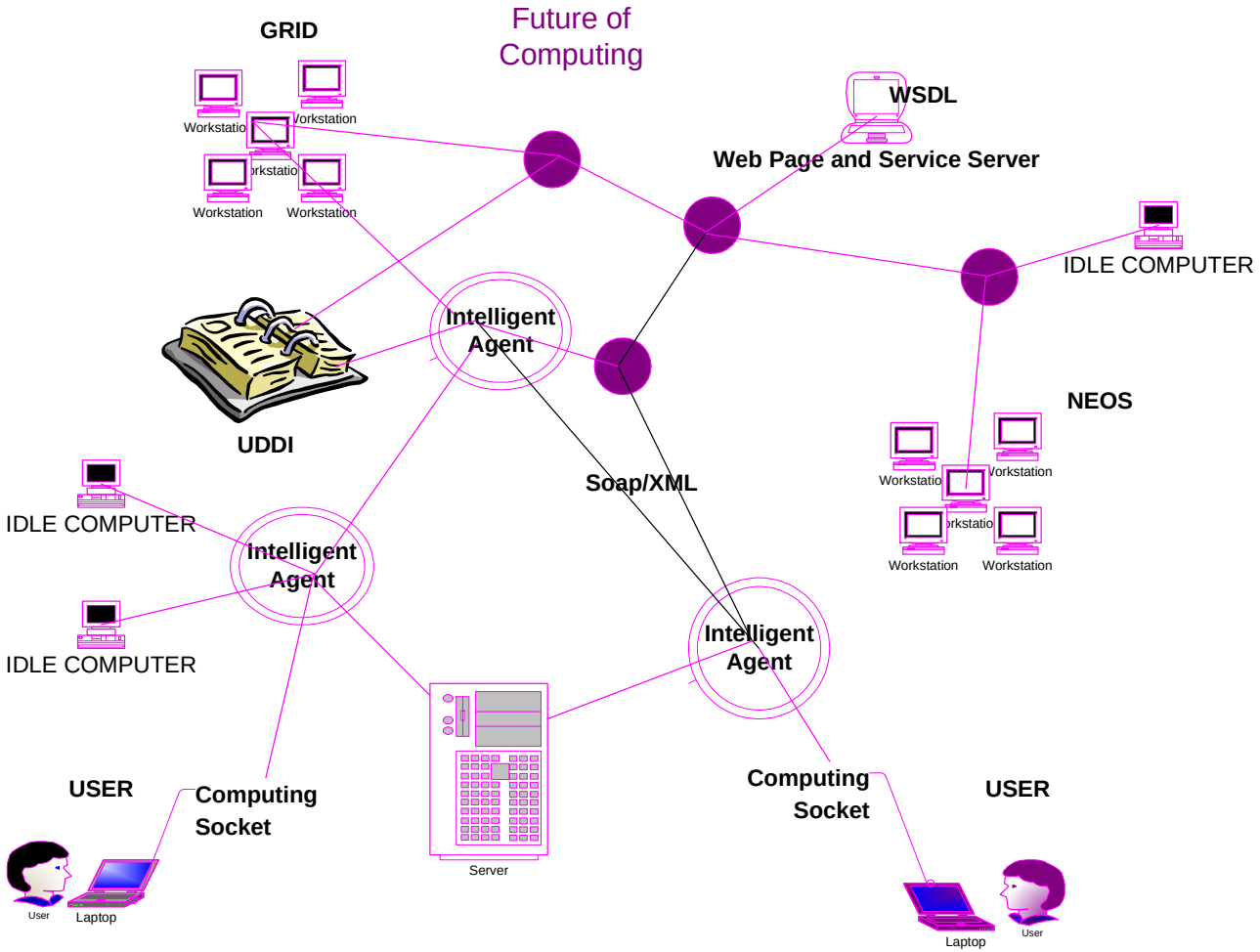
- There exist many heterogeneous implementations of optimization solvers (about 70)
- There exist many formats to represent optimization problems (about 10)
- There exist many mechanisms to communicate with solvers (about 10)
- There are some initiatives to analyze and benchmark optimization (?)
- Other Problems (All these possible reasons for recent NEOS leveling off, thus the motivation)
  - Different Operating Systems (about 5)
  - Different Programming Languages (about 5)
  - Locating solvers (none)
  - Reliability
  - Scalability
  - Security
- We propose a general design and framework
- We attempt to unify some of the popular means of solver communication, optimization analyses and model representations under the framework
- This can also be regarded an initiative to start a wider level of cooperation to move toward a final standardization and facilitate a healthier development environment for research in the area of operations research

Outline

- Abstract
- Background
  - A General Picture – The Future of Computing
  - Our Positioning – The Hierarchy of Operations Research
  - Some Terminologies (Parallel/Distributed/Grid Computing, XML, Web Services, etc.)
- Motivation
  - Motorola's VP Intelligent Optimization System
  - AMPL-NEOS
- The Design and Framework for Distributed Optimization (NSF Funded)
  - Proposing and Defining Optimization Services (OS, OSXL)
  - Design of Distributed Optimization Architecture (Centralized and Decentralized)
  - Optimization Services Representation (OSTL, OSRL, OSOL, OSSL, OSAL)
  - Optimization Services Communication (OSCL, OSDL, OSFL, OSEL)
  - Optimization Services Inspection and Discovery (OSIL, OSPL, OSBL, OSQL)
- Future Work and Conclusion

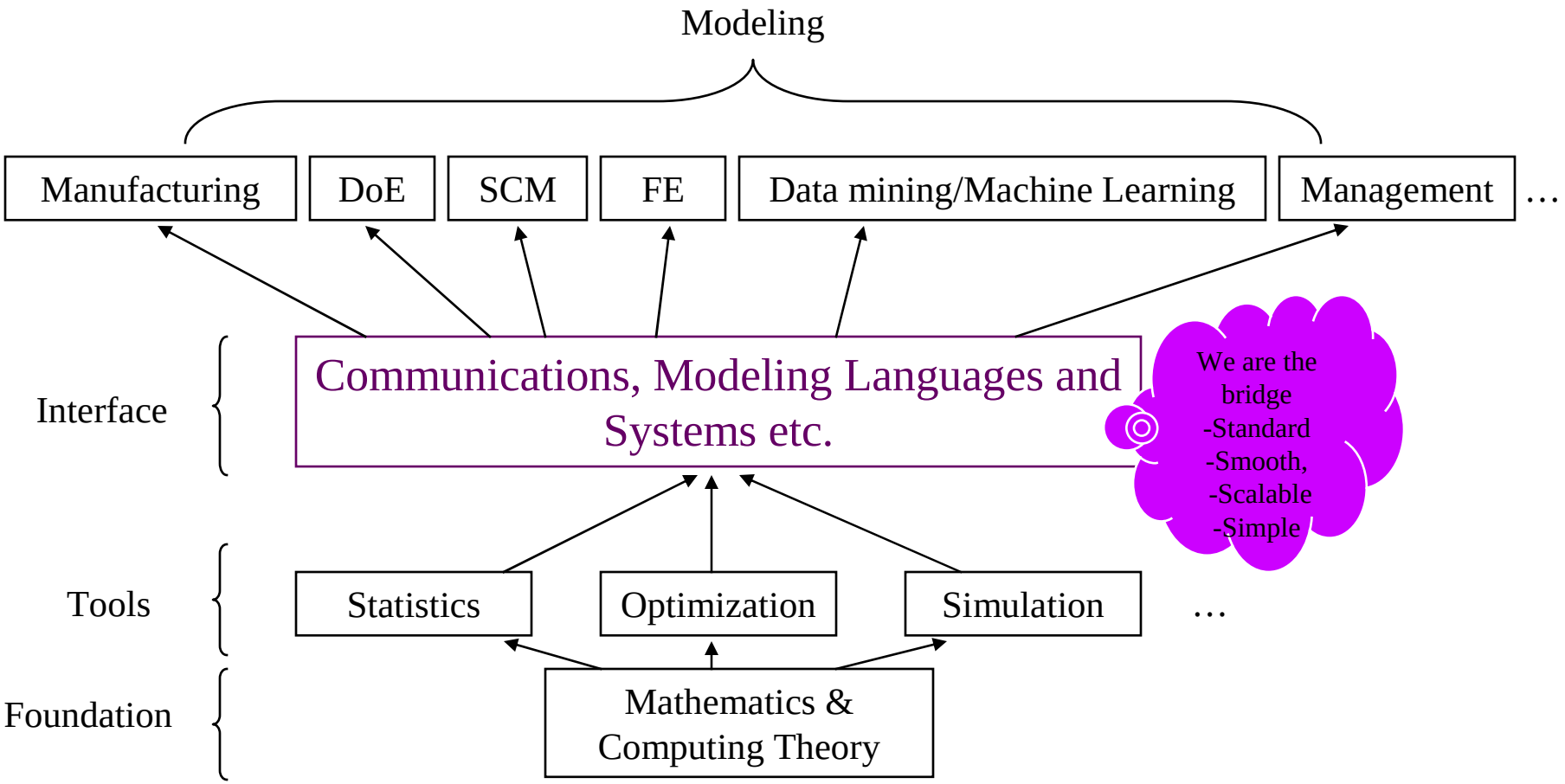
NORTHWESTERN UNIVERSITY

A General Picture



NORTHWESTERN UNIVERSITY

Our Positioning in Industrial Engineering and Management Sciences



Literature Reviews

- General Technologies
  - Web Services–latest technology that we leverage On [W3C, T. Berners-Lee etc.]
  - Open Grid Services Architecture [I. Foster, C. Kesselman]
  - Optimization as an Internet Resource [R. Fourer, J.-P. Goux]
- Modeling Systems
  - Motorola Virtual Prototyping System [T. Tirpak etc.]
  - NEOS – Network Enabled Optimization System [E.D. Dolan, R. Fourer, J.J. Moré, T.S. Munson, etc.]
- Optimization Representations
  - AMPL: A Modeling Language for Mathematical Programming [R. Fourer, D. Gay, B. Kernighan]
  - A W3C XML Schema for Linear Programming [R. Fouer, L. Lopez, K. Martin]
  - SMPS Standard [J. R. Birge, H. Gassmann, E. Gunn, A. King, and M. Dempster]
  - No widely adopted nonlinear extensions to Mathematical Programming System (MPS)
- Potential Applications and Collaborations
  - Integrated DEsign Automation Laboratory (IDEAL) [W. Chen]
  - COIN-OR project for publications [www-124.ib.com/developerworks/opensource/coin]
  - Optimization Methods on Computational Grids [MetaNEOS Prject, J. Linderoth, Lehigh U.]
- Other references throughout the presentation

Terminologies – Parallel/Distributed/Grid Computing

- Parallel Computing

  "Process by which a problem is solved using multiple resources working concurrently and collaboratively." [Class Notes on Parallel Computing, P. Banerjee]

- Distributed Computing

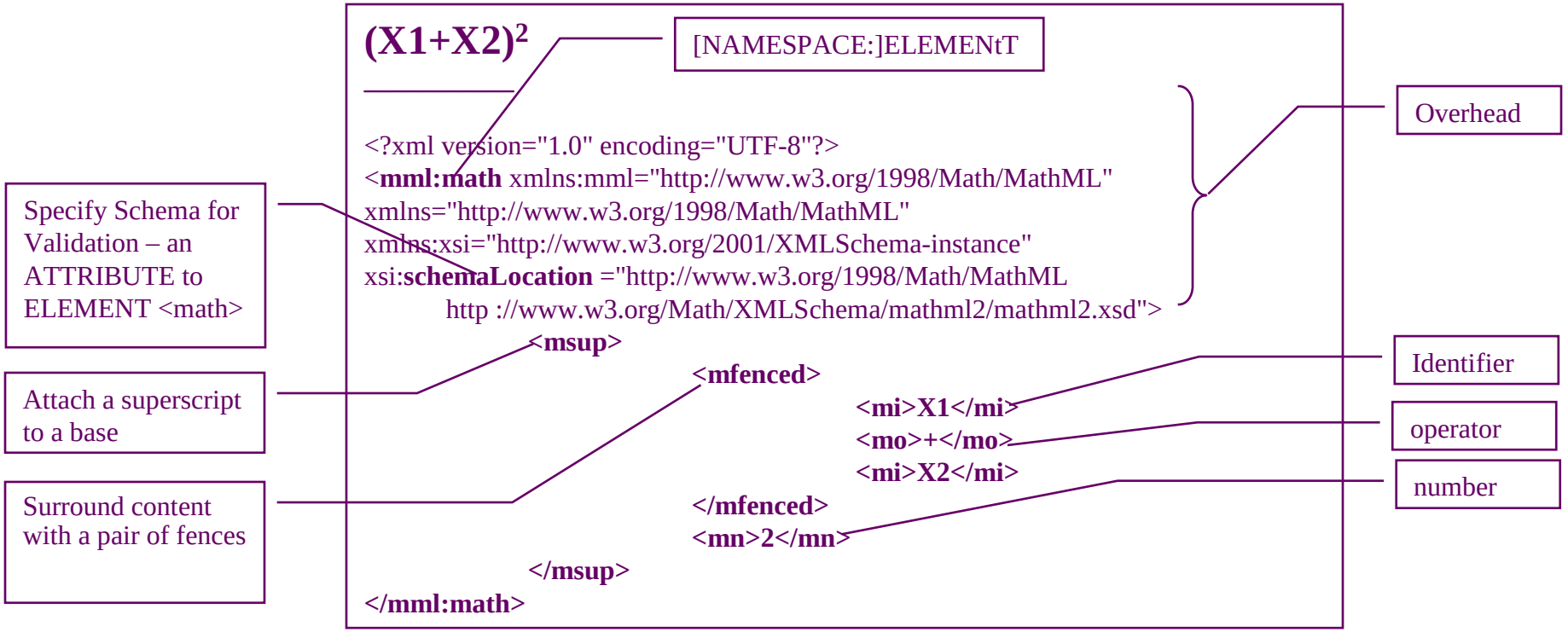
  "Computing on networked computers which is deeply concerned with problems such as reliability, security, and heterogeneity that are generally regarded as tangential in parallel computing." [Designing and Building Parallel Programs, I. Foster]

- Grid Computing

  "An ambitious and exciting global effort to develop an environment in which individual users can access computers, databases and experimental facilities simply and transparently, without having to consider where those facilities are located." [RealityGrid, Engineering & Physical Sciences Research Council, UK 2001]

- eXtensible Markup Language. A subset of SGML constituting a particular text markup language for interchange of structured data. It is a trademark of the World Wide Web Consortium.

- MathML – a dialect of XML – more for presentation than for computation)



XML Schema

- A database-inspired method for specifying constraints on XML documents using an XML-based language
- MathML Schema for element <msup>

```
<xs:complexType name="msup.type">
        <xs:group ref="Presentation-expr.class" minOccurs="2" maxOccurs="2" />
        <xs:attributeGroup ref="msup.attlist" />
</xs:complexType>

<xs:element name="msup" type="msup.type" />
```

```
<msup>
   <mfenced>
        <mi>X1</mi>
        <mo>+</mo>
        <mi>X2</mi>
   </mfenced>
   <mn>2</mn>
</msup>
```

Other XML Technologies

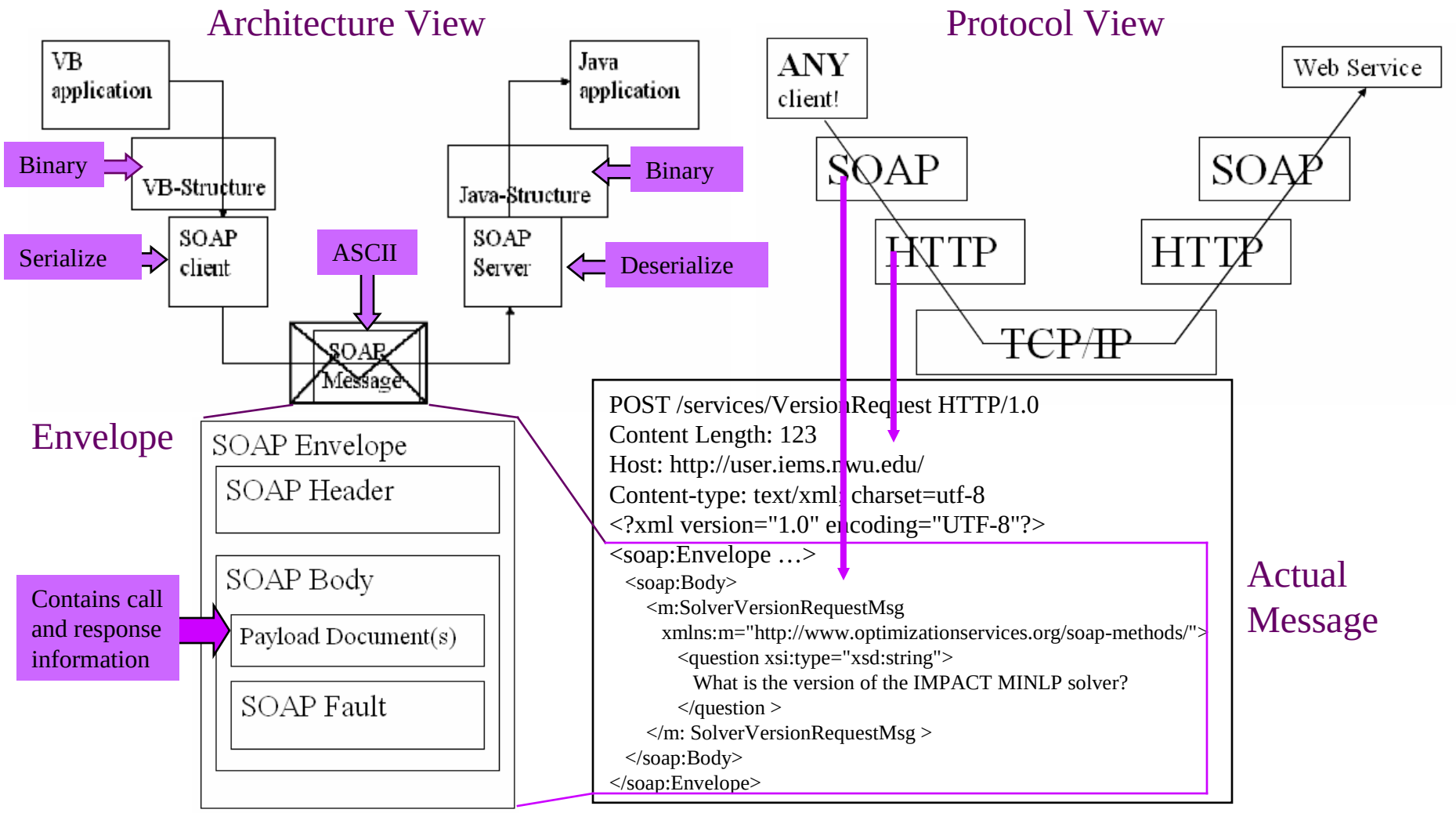
- XML **authoring**
  - XML **editors**
  - XML **validators**
- XML **transformation**
  - Tools for transforming XML into something that can be displayed in a browser or other rendering device.
  - **XSL,** and its associated language **XSLT,** is the main tool here.
- XML processing
  - **Xerces**
  - **XT**
- XML Parsing Model
  - **DOM**
  - **SAX**
- **Xpath –** for identifying subsets of a document, used in XSL , Xquery etc.
- **Xquery –** language for querying XML
- **Xlink, Xpointer–** for linking, referencing information within documents
- **XML Namespace** – tagged in front of element name to avoid potential name conflict.

Web Services & Simple Object Access Protocol

- Web Services – Platform and implementation independent components
  - **Described** using a service description language
  - **Published** to a registry of services
  - **Discovered** through a standard mechanism (at runtime or design time)
  - **Invoked** through a declared API, usually over a network
  - **Composed** with other services

- SOAP (1.2v, 24 June 2003) – first version that is a W3C recommendation
  - Lightweight protocol and platform independent
  - For exchange of information in a decentralized, distributed environment.
  - XML based
    - An **envelope** of message description and processing,
    - **Encoding rules** for data types
    - A convention for representing remote procedure **calls and responses**
    - A **binding** convention for exchanging messages using an underlying protocol.

Web Services & Simple Object Access Protocol



Web Services & Web Service Definition Language

- The standard format for describing a web service.
  - Describes how to access a web service and what operations it will perform
  - Can be provided through UDDI registries
  - XML based
    - PortType – Application Programming Interface
    - Operation -- Methods/Function prototype
    - Message – Parameters which can be input, output or fault message
    - Types -- Actual data types in a message, e.g. array of integers
    - Binding – Transport protocol and encoding protocol
    - Port – Network address
    - Service – A collection of ports

Web Services & Web Service Definition Language

```
<wsdl:definitions …>
    <message name="favoriteSolverRequest">
            <part name="question" type="xsd:string" />
    </message>
     <message name="favoriteSolverResponse">
            <part name="answer" type="xsd:string" />
    </message>
    <portType name="SimpleSolver">
            <operation name="favoriteSolver" parameterOrder="question">
               <input message="favoriteSolverRequest" …>
               <output message="favoriteSolverResponse" …>
            </operation>
    </portType>
    <binding name="SimpleSolverSoapBinding" type="intf:SimpleSolver">
            <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
            …
    </binding>
    <service name="SimpleSolverService">
            <port binding="intf:SimpleSolverSoapBinding" name="SimpleSolver">
                <wsdlsoap:address location="http://localhost:8080/axis/SimpleSolver.jws" />
            </port>
    </service>
</wsdl:definitions>
```

Interface

Protocol

Address

Name of Program

Method

Input & Output

Where the Web Service is

Web Services, Web Service Inspection Language & Universal Description Discovery and Integration

- WSIL
  - Specification for online web service registries (cf. web page search engines)
  - Light weighted and complements UDDI
  - XML based

```
<inspection …>
    <abstract>Impact is an Integrated Mathematical Programming Advanced Computational Tool.</abstract>
    <service>
            <name>Impact Solver Service</name>
            <abstract>The version of the Impact service is 1.0. It solves many types of optimization problems.</abstract>
            <description
                    referencedNamespace=http://schemas.xmlsoap.org/wsdl/
                    location="http://www.optimizationservices.org/os/ossolver/ImpactSolverService?wsdl">
            </description>
    </service>
    <link location=" http://www.optimizationservices.org/os/ossolver/JunMaSolverService.wsil" >
            <abstract>JunMa Solver Service</abstract>
    </link>
</inspection>
```

- UDDI
  - Heavyweight (think of yellow pages)
  - Also specifies business models
  - Provides a complete set of SOAP query APIs for searching web services

Open Grid Services Architecture & Web Services

- OGSA (Globus Toolkit)
  - Protocol deficiencies
  - Mainly in scientific world
  - Missing functionality
  + Support transient services
  + Reliable and secure transport
  + Service creation
  + Global naming and references
  + Service management
  + Registration
  + Notification
  + Authorization
  + Concurrency

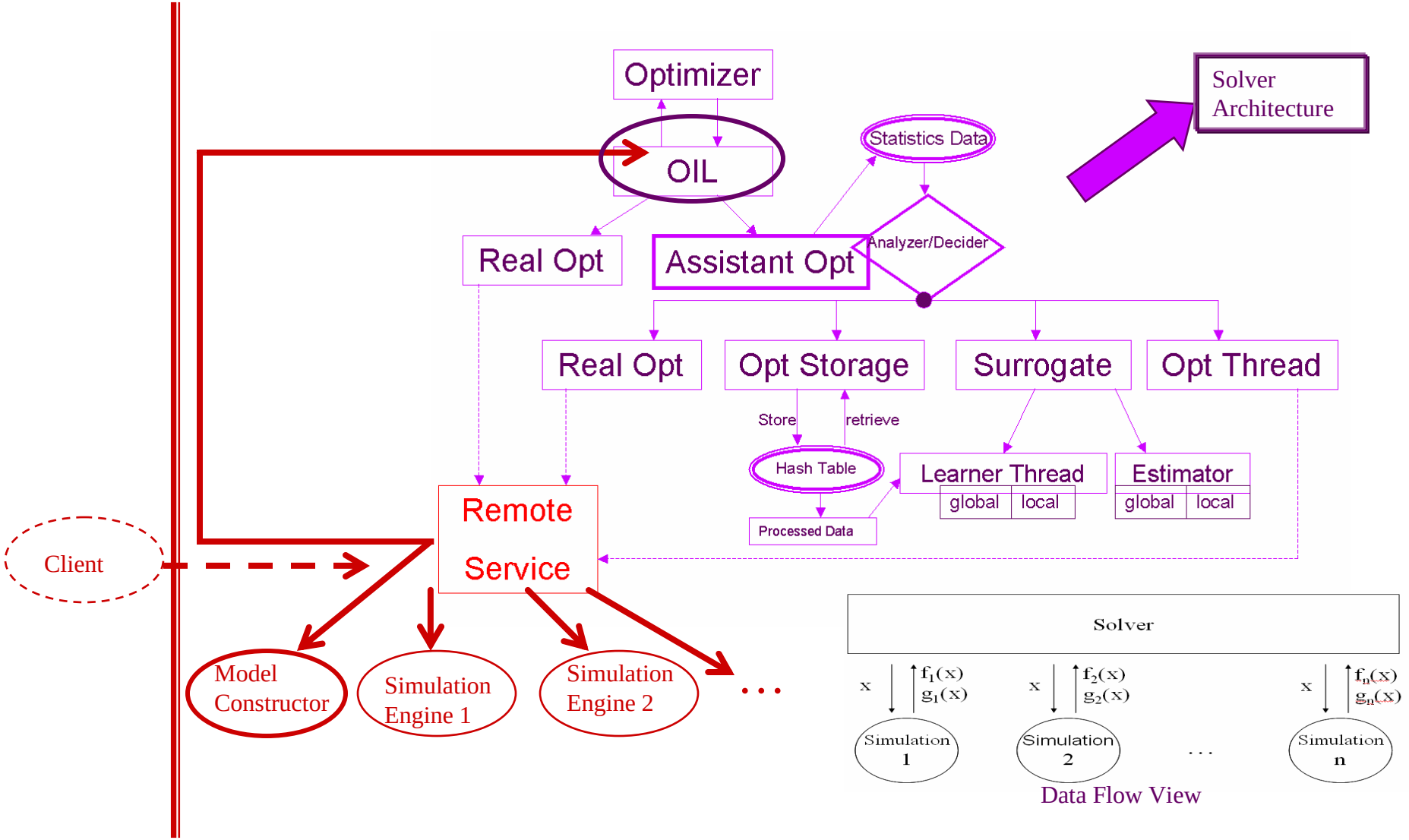
- Web Service
  + Increasingly popular
  + Standards-based

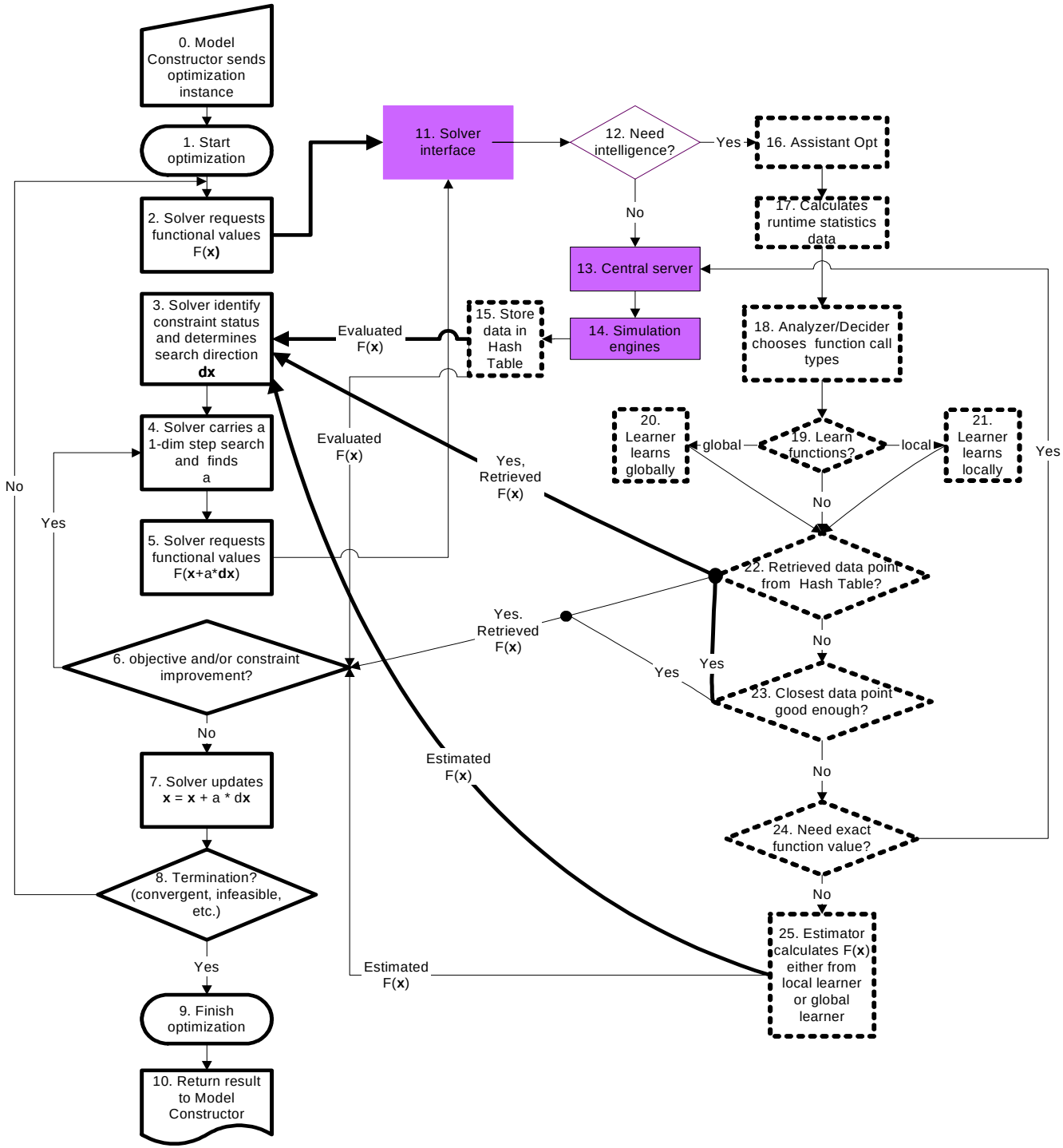
- Web services w/ grid – good idea
- Becoming a topic at the super computing conferences
- Will eventually converge till no distinction
  - My feeling, and my hope
  - Not a question of who wins
  - Both provide something
- Good for our Optimization Services

Outline

- Abstract
- Background
  - A General Picture – The Future of Computing
  - Our Positioning – The Hierarchy of Operations Research
  - Some Terminologies (Parallel/Distributed/Grid Computing, XML, Web Services, etc.)
- Motivation
  - Motorola's VP Intelligent Optimization System
  - AMPL-NEOS
- The Design and Framework for Distributed Optimization (NSF Funded)
  - Proposing and Defining Optimization Services (OS, OSXL)
  - Design of Distributed Optimization Architecture (Centralized and Decentralized)
  - Optimization Services Representation (OSTL, OSRL, OSOL, OSSL, OSAL)
  - Optimization Services Communication (OSCL, OSDL, OSFL, OSEL)
  - Optimization Services Inspection and Discovery (OSIL, OSPL, OSBL, OSQL)
- Future Work and Conclusion

Motorola's VP Intelligent Optimization System Architect and Procedures




0. Model Constructor sends optimization instance
1. Start optimization
2. Solver requests functional values F(x)
3. Solver identify constraint status and determines search direction dx
4. Solver carries a 1-dim step search and finds a
5. Solver requests functional values F(x+a*dx)
6. objective and/or constraint improvement?
7. Solver updates x = x + a * dx
8. Termination? (convergent, infeasible, etc.)
9. Finish optimization
10. Return result to Model Constructor
11. Solver interface
12. Need intelligence?
13. Central server
14. Simulation engines
15. Store data in Hash Table
16. Assistant Opt
17. Calculates runtime statistics data
18. Analyzer/Decider chooses function call types
19. Learn functions?
20. Learner learns globally
21. Learner learns locally
22. Retrieved data point from Hash Table?
23. Closest data point good enough?
24. Need exact function value?
25. Estimator calculates F(x) either from local learner or global learner
Yes
No
Evaluated F(x)
Yes, Retrieved F(x)
Yes. Retrieved F(x)
Estimated F(x)
global
local

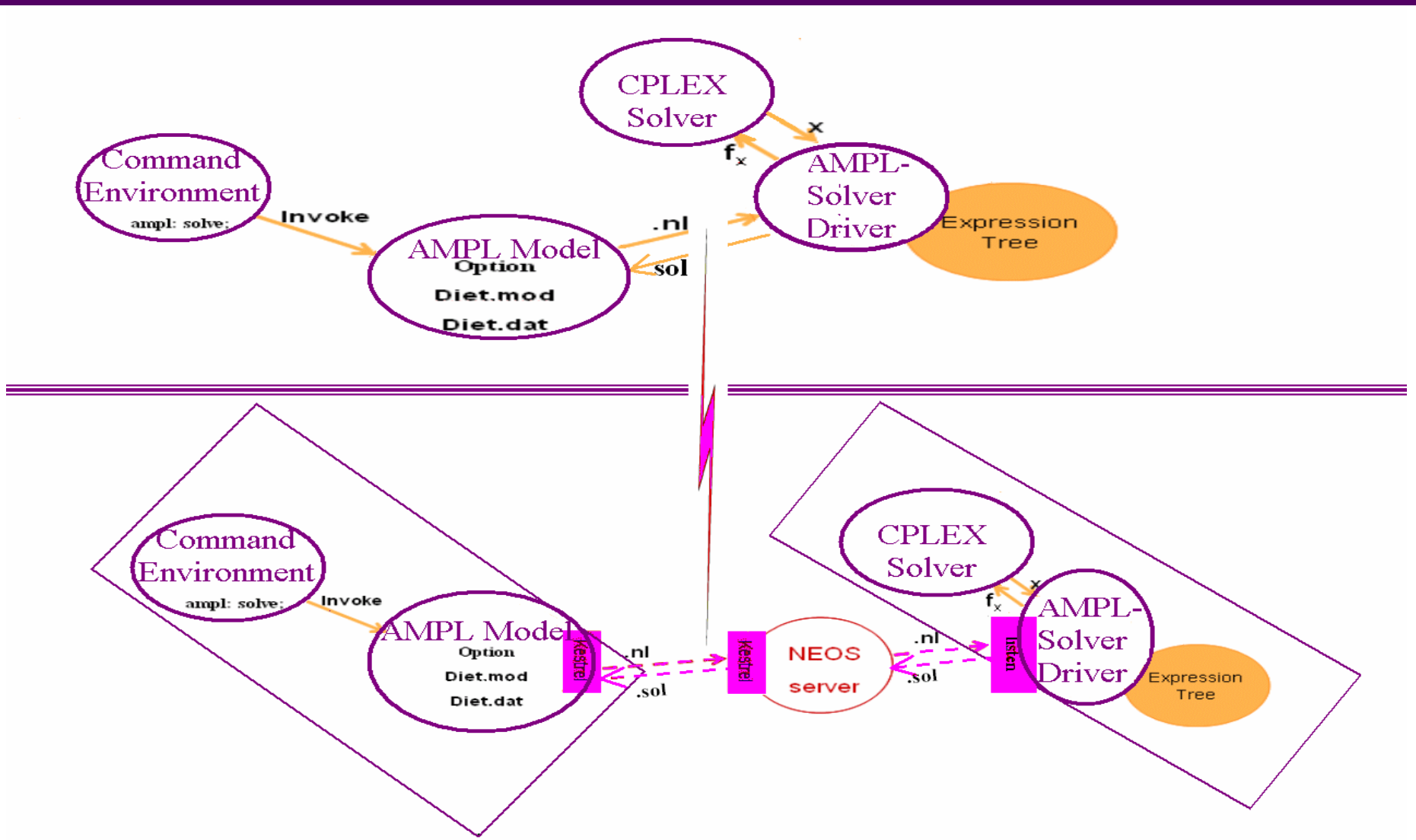
Motorola's VP Intelligent Optimization System Service Requirement

+ System Architecting
+ Interfacing & Invocation
    - Client & Server
    - Model & Solver
    - Solver & Simulation
+ Data Flow and Representation
+ Input Parsing and Output Generation
+ Variable/Objective/Constraint Construction
+ Service Location & Discovery (Not a problem in a centralized ASP)
? Common Variable Resolution (Can be included in OSFL)
? Simulation Service Flow (Can be included in OSFL)
? Metrics Interpretation (To a certain degree given naming standardization)
? Process Monitoring & Management (OSEL, OSPL)
? Job Queue Management (OSEL)
? Interdisciplinary Domain Knowledge Management (OSFL)

- Result Presentation/Mapping
- Intelligence Learning (Local Surrogate/Accelerator)

("+" intended to solve in our framework; "?" Partially supported; "–" Not in our framework domain)

AMPL - NEOS


CPLEX Solver
x
f_x
AMPL-Solver Driver
Expression Tree
Command Environment
ampl: solve;
Invoke
AMPL Model
Option
Diet.mod
Diet.dat
.nl
sol
Command Environment
ampl: solve;
Invoke
AMPL Model
Option
Diet.mod
Diet.dat
Kestrel
.nl
.sol
Kestrel
NEOS server
.nl
.sol
CPLEX Solver
x
f_x
Listen
AMPL-Solver Driver
Expression Tree

Outline

- Abstract
- Background
  - A General Picture – The Future of Computing
  - Our Positioning – The Hierarchy of Operations Research
  - Some Terminologies (Parallel/Distributed/Grid Computing, XML, Web Services, etc.)
- Motivation
  - Motorola's VP Intelligent Optimization System
  - AMPL-NEOS
- The Design and Framework for Distributed Optimization (NSF Funded)
  - Proposing and Defining Optimization Services (OS, OSXL)
  - Design of Distributed Optimization Architecture (Centralized and Decentralized)
  - Optimization Services Representation (OSTL, OSRL, OSOL, OSSL, OSAL)
  - Optimization Services Communication (OSCL, OSDL, OSFL, OSEL)
  - Optimization Services Inspection and Discovery (OSIL, OSPL, OSBL, OSQL)
- Future Work and Conclusion

Proposing ”Optimization Services”

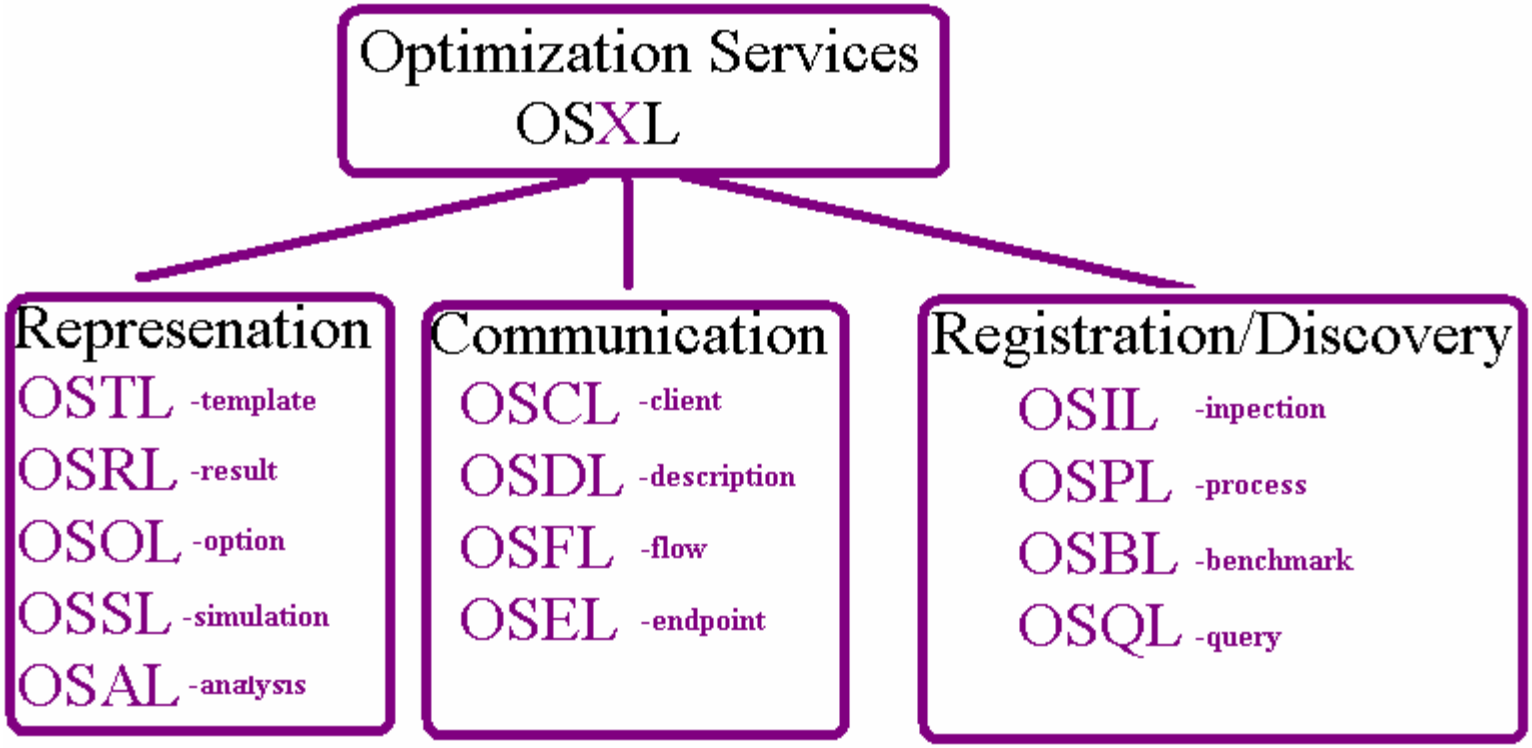
- Optimization Services (temporary definition)
  - SOAP based web services (potentially also leveraging on grid computing technologies) with specified interfaces and behaviors under the general framework of distributed optimization, including the following OSXL’s:

    for representing optimization instances,
    - Optimization Services Template Language (OSTL)
    - Optimization Services Result Language (OSRL)
    - Optimization Services Option Language (OSOL)
    - Optimization Services Simulation Language (OSSL)
    - Optimization Services Analysis Language (OSAL)

    for controlling optimization accesses, flows and operations,
    - Optimization Services Client Language (OSCL)
    - Optimization Services Description Language (OSDL)
    - Optimization Services Flow Language (OSFL)
    - Optimization Services Endpoint Language (OSEL)

    for discovering and inspecting optimization services,
    - Optimization Services Inspection Language (OSIL)
    - Optimization Services Process Language (OSPL)
    - Optimization Services Benchmark Language (OSBL)
    - Optimization Services Query Language (OSQL)

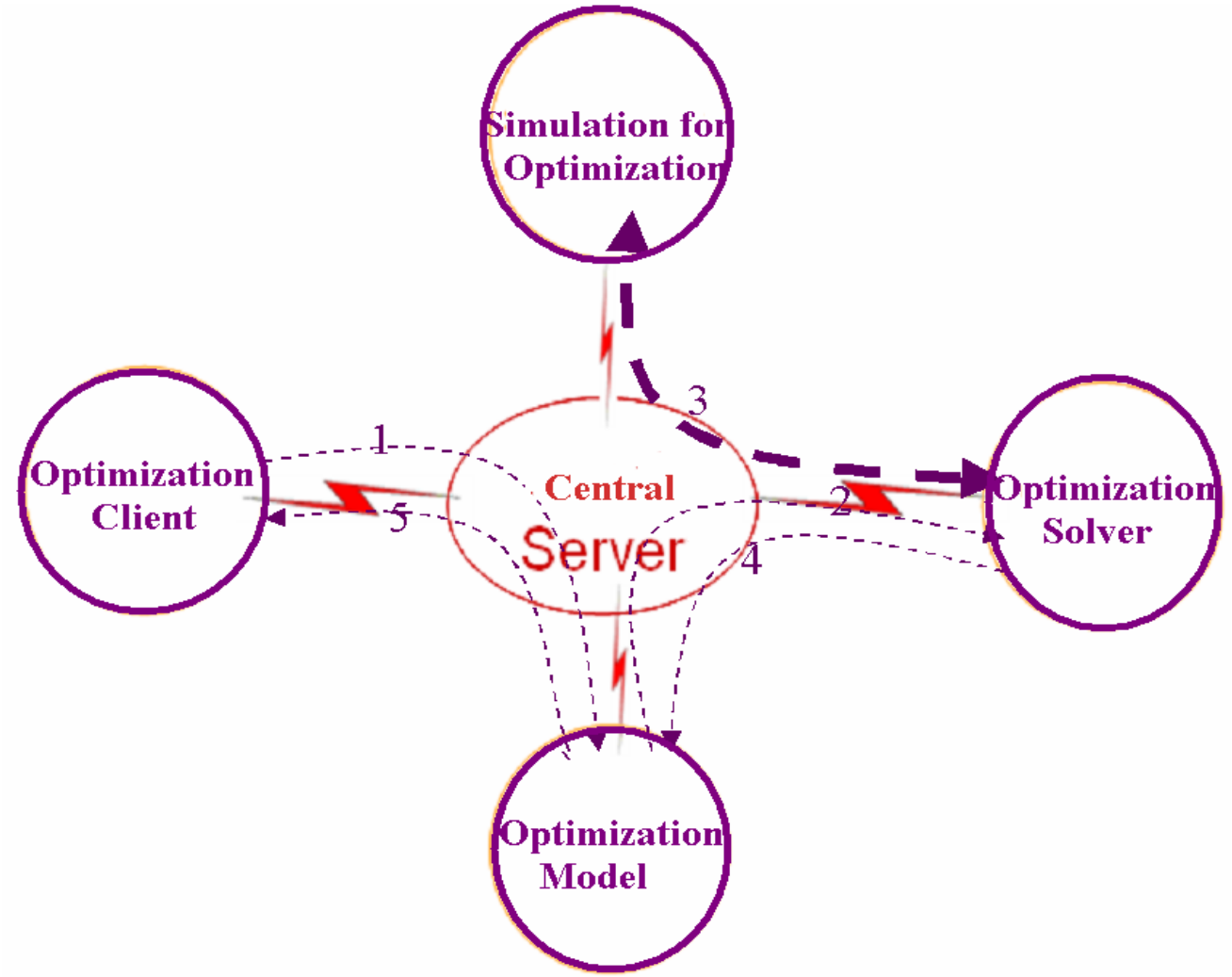
Proposing ”Optimization Services”



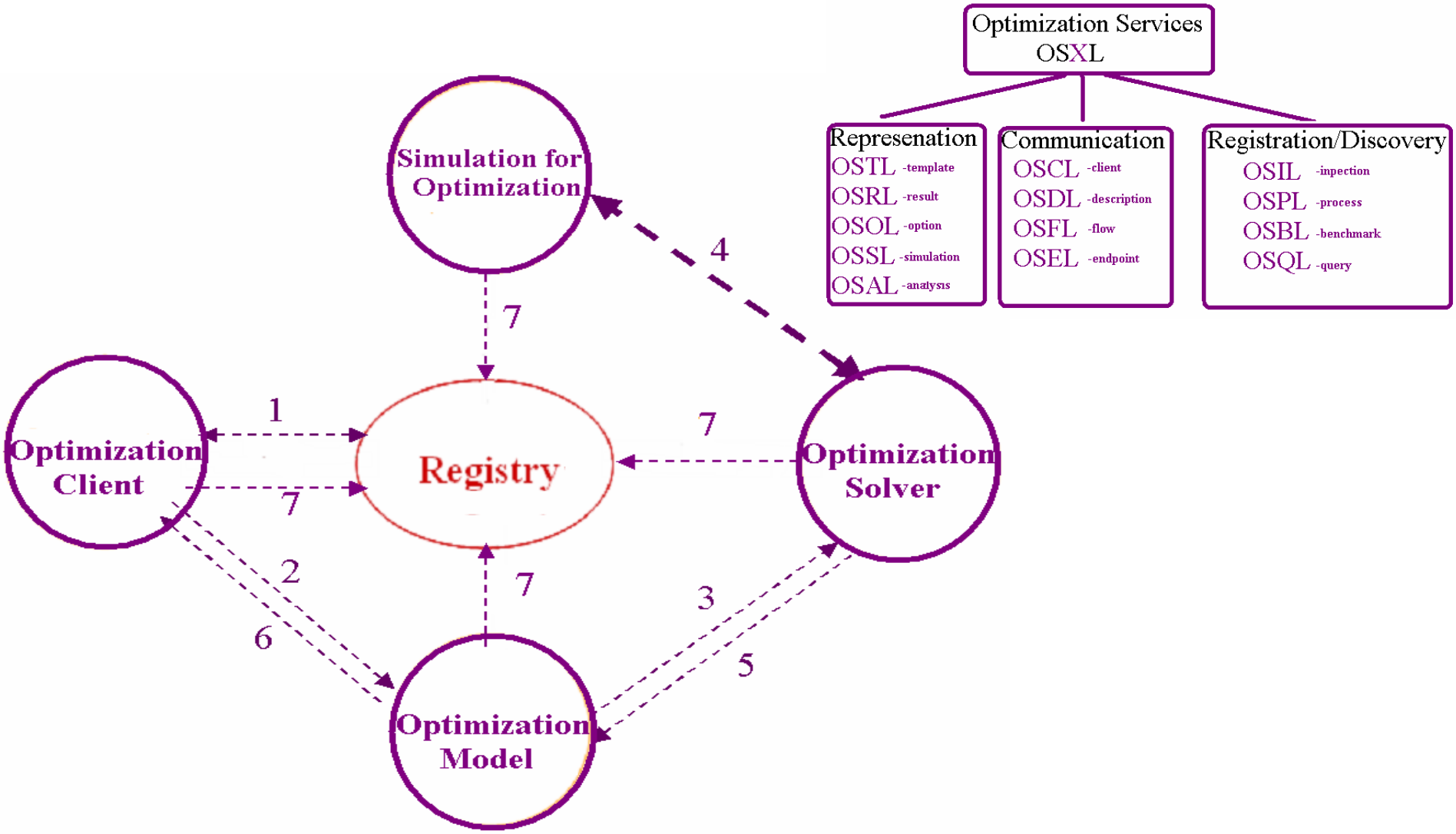
Outline

- Abstract
- Background
  - A General Picture – The Future of Computing
  - Our Positioning – The Hierarchy of Operations Research
  - Some Terminologies (Parallel/Distributed/Grid Computing, XML, Web Services, etc.)
- Motivation
  - Motorola's VP Intelligent Optimization System
  - AMPL-NEOS
- The Design and Framework for Distributed Optimization (NSF Funded)
  - Proposing and Defining Optimization Services (OS, OSXL)
  - Design of Distributed Optimization Architecture (Centralized and Decentralized)
  - Optimization Services Representation (OSTL, OSRL, OSOL, OSSL, OSAL)
  - Optimization Services Communication (OSCL, OSDL, OSFL, OSEL)
  - Optimization Services Inspection and Discovery (OSIL, OSPL, OSBL, OSQL)
- Future Work and Conclusion

Design of Distributed Optimization Architect Centralized

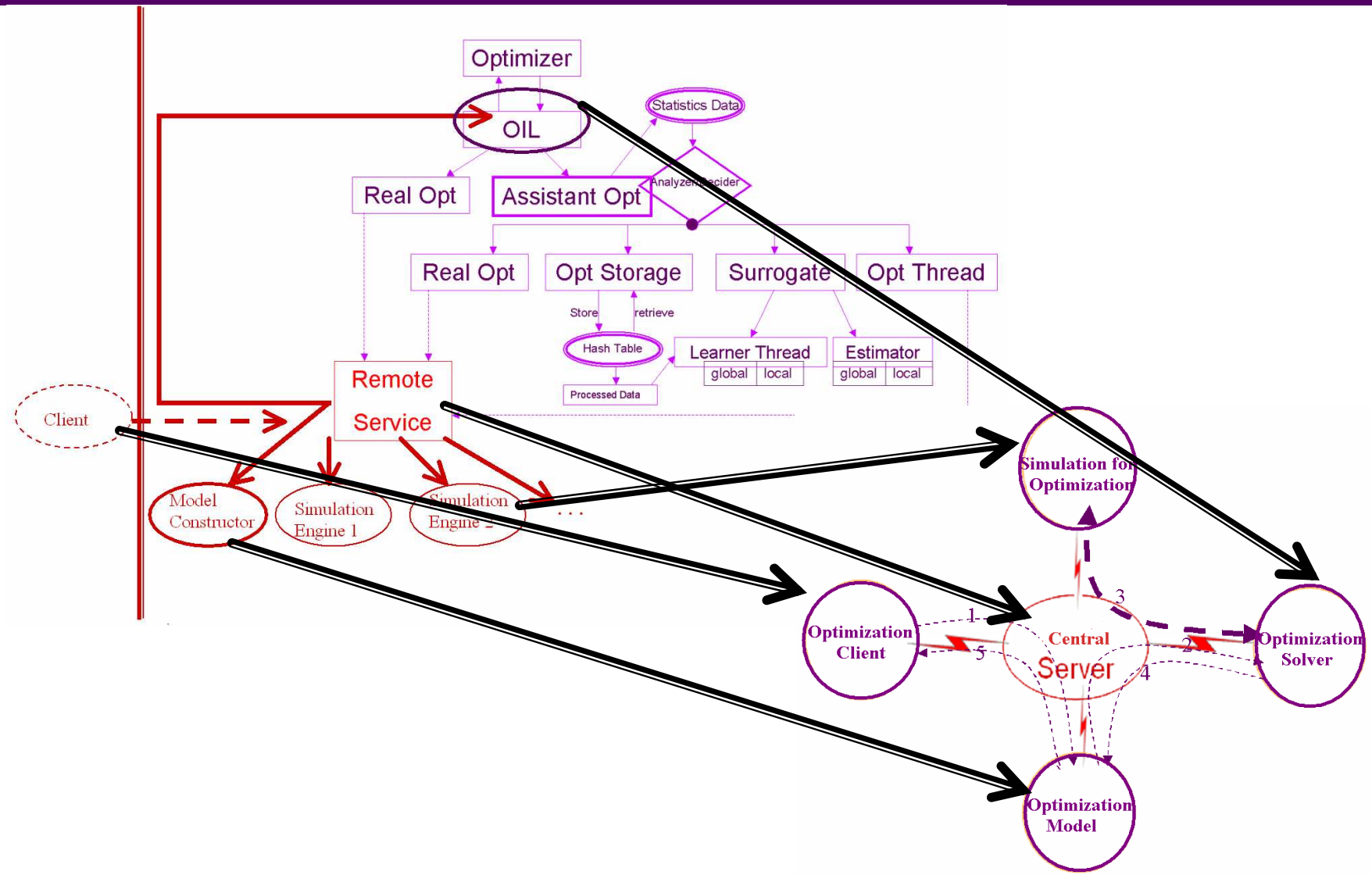


Design of Distributed Optimization Architect Decentralized (Recommended)

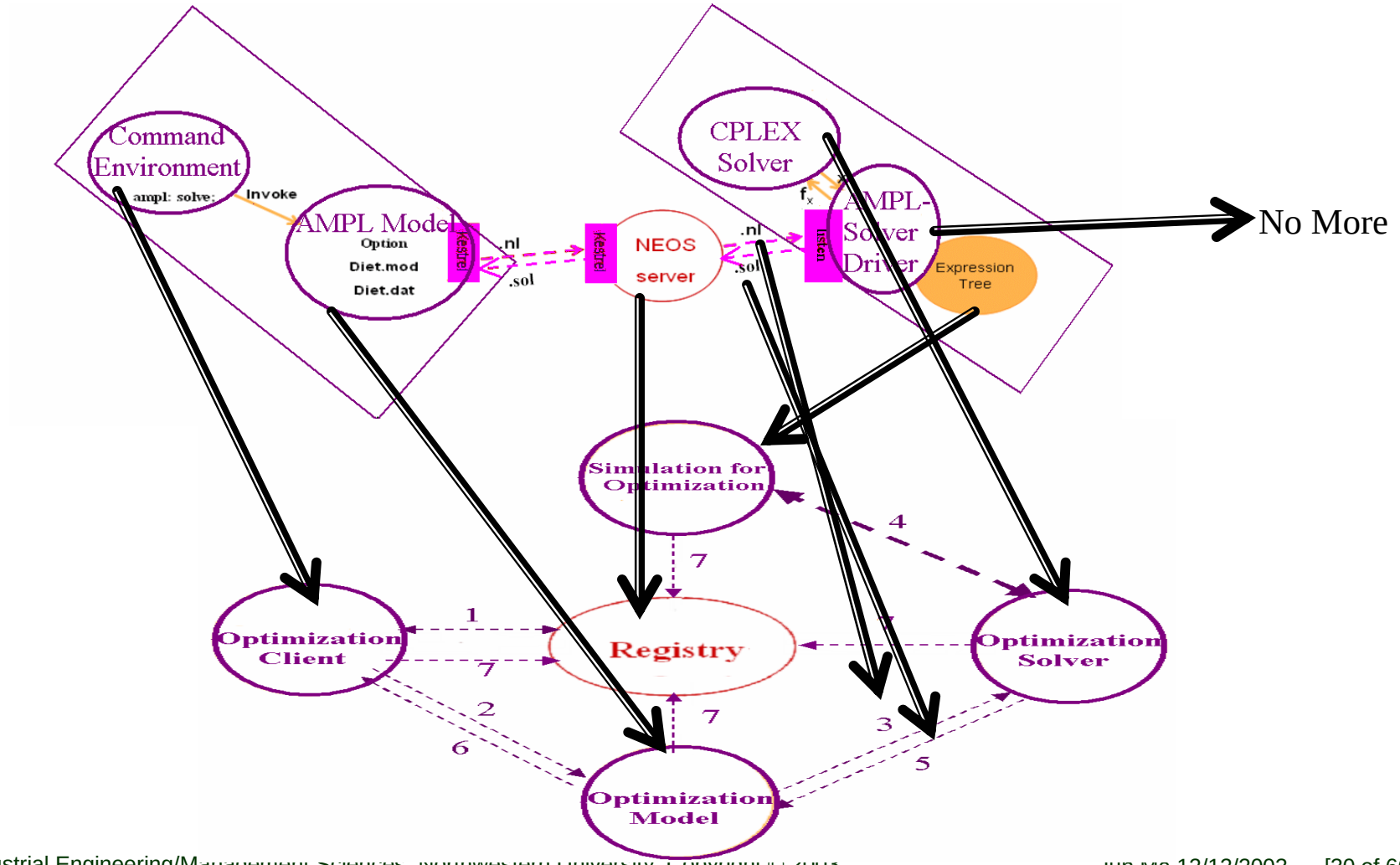


NORTHWESTERN UNIVERSITY

Application Service Provider Revisited



AMPL-NEOS Revisited



Outline

- Abstract
- Background
  - A General Picture – The Future of Computing
  - Our Positioning – The Hierarchy of Operations Research
  - Some Terminologies (Parallel/Distributed/Grid Computing, XML, Web Services, etc.)
- Motivation
  - Motorola's VP Intelligent Optimization System
  - AMPL-NEOS
- The Design and Framework for Distributed Optimization (NSF Funded)
  - Proposing and Defining Optimization Services (OS, OSXL)
  - Design of Distributed Optimization Architecture (Centralized and Decentralized)
  - Optimization Services Representation (OSTL, OSRL, OSOL, OSSL, OSAL)
  - Optimization Services Communication (OSCL, OSDL, OSFL, OSEL)
  - Optimization Services Inspection and Discovery (OSIL, OSPL, OSBL, OSQL)
- Future Work and Conclusion

High and Low Levels of Optimization Representation

- High Level Optimization Representation
  - Example: AMPL, GAMS, LINGO, ILOG OPL etc.
  - Abstract algebraic representation of a problem
  - Separation between model and data
- Optimization Services Representation
  - Example: MPS – linear programming, SMPS – stochastic programming, .nl – used in AMPL, numerous proprietary formats used in commercial solvers
  - Considered in our framework
  - An optimization problem instance
  - Generated by filling the model with corresponding data

Lower Level Optimization Representation

Optimization Services Template Language (OSTL)

- It's not "Optimization Services Modeling language." It is "Optical Spectroscopy Markup Language" (The National Scientific Research Center – France)
- "Modeling" may be confused with high level algebraic modeling languages.
- A unified optimization instance representation – think of skeleton or simply template
- Possible function type to be included (listed in terms of possibility)
  - Existing standard linear representation in XML
  - Simulation engine used for function evaluation through optimization
    - Called through OSCL
    - Represented through OSSL
  - MathML for simple relation and nonlinear representation
  - Existing standard linear representation in ASCII
  - Binary expression or function calculation executable that takes OSSL as input and output
- Constraints expressed as combination of individual constraints (mainly nonlinear or closed form) and constraint sets (mainly for linear form and matrices)
- Can contain an optional section of optimization analysis represented by OSAL

Lower Level Optimization Representation

Optimization Services Template Language (OSTL)

```
<OSML>
    <singleFomrat type="...">
        <!--anyformat-->
    </singleFomrat>
</OSML>
```

or

```
<OSML>
    <mixedFormat>
        <variables>
            <variable name="X1" type="countinuous" lowerBound="1" upperBound="3">1.2</variable>
            <variable name="X2" type="binary">0</variable>
            <variable name="X3" type="integer"></variable>
        </variables>
        <objective name="totoalCost">
            <direction>minimize</direction>
            <function>
                <webService>
                    <URI>
                    </URI>
                    <OSSL>
                    </OSSL>
                </webService>
            </function>
            <lowerBound>-100</lowerBound>
            <upperBound>-100</upperBound>
        </objective>
```

NORTHWESTERN UNIVERSITY

Lower Level Optimization Representation

Optimization Services Template Language (OSTL)

```
<constraints>
    <constraint name="c1">
        <!--for expressing nonlinear constraints-->
        <function>
            <binary language="java" platform="unix">
                <URI>
                </URI>
                <OSSL>
                </OSSL>
            </binary>
        </function>
        <lowerBound>-3</lowerBound>
        <upperBound>5</upperBound>
    </constraint>
    <constraint name="c2">
        <function>
            <MathML>
                <!--terms can be functions, which in term can be binary, webservice, or mathml, besides values and defined variables-->
            </MathML>
        </function>
        <upperBound>5</upperBound>
    </constraint>
    <constraintSet name="cset" type="MPS">
        <!--for expressing linear constraints-->
    </constraintSet>
</constraints>
</mixedFormat>
<OSAL>
    <!--analysis data-->
</OSAL>
</OSML>
```

Lower Level Optimization Representation

Optimization Services Result Language (OSRL)

- Contains 4 sections
  - Status
  - Variables
  - Objective
  - Constraints
  - May contain extended contents
- Names of different types of results associated with variables, the objective and constraints should be standardized (don't necessarily have to be reported)
- Think of AMPL suffixes

Lower Level Optimization Representation

Optimization Services Result Language (OSRL)

```
<OSRL>
  <status>
    <!--unbounded, found, infeasible, error-->
  </status>
  <variables>
    <variable name="x1">
      <standard>
        <R name="value">12</R>
      </standard>
      <specific>
        <R name="weirdValue">
          <description>some weird values only calculated by this solver</description>
        </R>
      </specific>
    </variable>
    <variable name="x2">
      ...
    </variable>
</variables>
```

```
  <objective name="totalCost">
    <standard>
      <R name="value">3</R>
    </standard>
    <specific>
      ...
    </specific>
  </objective>
  <constraints>
    <constraint name="c1">
      <standard>
        <R name="body">12</R>
      </standard>
      <specific>
        ...
      </specific>
    </constraint>
    <constraint name="c2">
      ...
    </constraint>
  </constraints>
</OSRL>
```

Lower Level Optimization Representation

Optimization Services Option Language (OSOL)

- Separate from OSTL, because it is solver specific, rather than problem specific
- Contains 2 sections
  - Standard
    - Common names of solver options
    - Contains optional description (because it is standardized)
    - Can be used for solver inspection and discovery
  - Specific
    - Solver specific
    - Better contains a description for each option
    - Option values are typeless (For simplification. Think of Python)

Lower Level Optimization Representation

Optimization Services Option Language (OSOL)

```
<OSOL>
    <standard>
        <!--standard can be used for inspection and discovery-->
        <O name="maxIter">
            <description>maximum number of iterations</description>
            1.0
        </O>
        <O name="maxTime">
            2.0
        </O>
    </standard>
    <specific>
        <!--solver specific-->
        <O name="weirdSolverOption">
            <description>...</description>
        abc
    </O>
        <O name="...">
            ...
    </O>
    </specific>
</OSOL>
```

Lower Level Optimization Representation

Optimization Services Simulation Language (OSSL)

- Used both for client initiating optimization model and solver calling simulation engines
- Contains an input and/or an output section.
- Input section contains two types of elements, both are typeless
  - Variable
  - Parameter
- From perspective of simulation engine or function evaluator (binary executables), they are both input arguments
- Reasoning for having two types of input argument
  - Parameters are fixed, whereas variables change
  - Iterative nature of optimization
  - Network trafficking
  - Variable matching between OSSL and OSML
  - Different treatment for numerical analysis, e.g. variable be represented more accurately for calculating derivatives

Lower Level Optimization Representation

Optimization Services Simulation Language (OSSL)

```
<OSSL>
    <Input>
        <param name="p1">23</param>
        <param name="p2">abc</param>
        <var name="X1">34</var>
        <var name="X2">44</var>
    </Input>
    <Output>
        <metrics name="m1">33</metrics>
        <metrics name="m1">abc</metrics>
    </Output>
</OSSL>
```

Lower Level Optimization Representation

Optimization Services Analysis Language (OSAL)

- Separate from OSTL, because it is analyzer specific, as well as problem specific
- AMPL .nl file does some analysis
- Analyzer [H.J. GreenBerg]
- MProbe [J.W. Chinneck, Carleton U.]
- Convexity detection [D. Orban, NU]
- Other solver recommendation projects [A. Neumeier, U. of Vienna]
- Contains 2 sections
  - Standard
    - Common names and analyses from analyzers
    - Contains optional description (because it is standardized)
  - Specific
    - Analyzer specific
    - Better contains a description for each option
    - Analysis values are typeless

Lower Level Optimization Representation

Optimization Services Analysis Language (OSAL)

```
<OSAL>
    <standard>
        <A name="OptimizationType">
            <description>describing optimization problems type</description>
            Nonlinearly Constrained Optimization
        </A>
        <A name="...">
            ...
        </A>
    </standard>
    <specific>
        <A name="weirdAnalysis">
            <description>...</description>
        abc
        </A>
        <A name="...">
            <description>...</description>
            ...
        </A>
    </specific>
</OSAL>
```

Outline

- Abstract
- Background
  - A General Picture – The Future of Computing
  - Our Positioning – The Hierarchy of Operations Research
  - Some Terminologies (Parallel/Distributed/Grid Computing, XML, Web Services, etc.)
- Motivation
  - Motorola's VP Intelligent Optimization System
  - AMPL-NEOS
- The Design and Framework for Distributed Optimization (NSF Funded)
  - Proposing and Defining Optimization Services (OS, OSXL)
  - Design of Distributed Optimization Architecture (Centralized and Decentralized)
  - Optimization Services Representation (OSTL, OSRL, OSOL, OSSL, OSAL)
  - Optimization Services Communication (OSCL, OSDL, OSFL, OSEL)
  - Optimization Services Inspection and Discovery (OSIL, OSPL, OSBL, OSQL)
- Future Work and Conclusion

Lower Level Optimization Communication

Optimization Services Client Language (OSCL)

- Specifies the following client interface for it to get function values

  String call (String input)
- Both return value and input value should be of the XML form specified in OSSL
- Default binding should be SOAP to HTTP
- Used to call a standard web service used as a simulation for optimization
- When solver needs a function value from a web service, the solver is considered a client
- It should be a Remote Procedure Call
- Port address (location of web service) should be specified

Lower Level Optimization Communication

Optimization Services Client Language (OSCL)

```
<OSCL:definitions xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <types/>
    <message name="callRequest">
        <part name="OSSLRequest" type="xsd:string"/>
    </message>
    <message name="callResponse">
        <part name="OSSLResponse" type="xsd:string"/>
    </message>
    <portType name="client">
        <operation name="call" parameterOrder="OSSLRequest">
            <input message="callRequest" name="callRequest"/>
            <output message="callResponse" name="callResponse"/>
        </operation>
    </portType>
```

```
    <binding name="clientSoapBinding" type="client">
        <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
        <operation name="call">
            <wsdlsoap:operation soapAction=""/>
            <input name="callRequest">
                <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </input>
            <output name="callResponse">
                <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="encoded"/>
            </output>
        </operation>
    </binding>
    <service name="clientService">
        <port binding="clientSoapBinding" name="client">
            <wsdlsoap:address location="http://localhost:8080/axis/client.jws"/>
        </port>
    </service>
</OSCL:definitions>
```

Lower Level Optimization Communication

Optimization Services Description Language (OSDL)

- Mechanism similar to OSCL
- Specifies the following 3 standard solver interfaces

  int solver (binary solve)
  - Input specifies whether just checking status (false) or finally need to solve (true)
  - Output reports solver status (possibly -1 if not ready, or a positive integer indicating job number for later retrieval)

  String set (String input)
  - Both input and output should be of the XML format specified in OSOL
  - Option values are set to empty if certain option input cannot be resolved

  String solve (String input)
  - Input should be of the XML format specified in OSTL
  - Output should be of the XML format specified in OSRL
- Other interfaces are possible but solver/optimization specific
- But probably don't need a stop function (We may well leverage it as well as other similar "management" functions on grid computing technologies. So hope they will be merged with web services)

Lower Level Optimization Communication

Optimization Services Description Language (OSDL)

```
<OSDL:definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <types/>
    <message name="solverRequest">
        <part name="jobRequest" type="xsd:boolean"/>
    </message>
    <message name="solverResponse">
        <part name="jobResponse" type="xsd:int"/>
    </message>
    <message name="setRequest">
        <part name="OSOLRequest" type="xsd:string"/>
    </message>
    <message name="setResponse">
        <part name="OSOLResponse" type="xsd:string"/>
    </message>
    <message name="solveRequest">
        <part name="OSMLRequest" type="xsd:string"/>
    </message>
    <message name="solveResponse">
        <part name="OSRLResponse" type="xsd:string"/>
    </message>
    <portType name="solver">
        <operation name="solver" parameterOrder="jobRequest">
            <input message="solverRequest" name="solverRequest"/>
            <output message="solverResponse" name="solverResponse"/>
        </operation>
        <operation name="set" parameterOrder="OSOLRequest">
            <input message="setRequest" name="setRequest"/>
            <output message="setResponse" name="setResponse"/>
        </operation>
        <operation name="solve" parameterOrder="OSMLRequest">
            <input message="solveRequest" name="solveRequest"/>
            <output message="solveResponse" name="solveResponse"/>
        </operation>
    </portType>
```

NORTHWESTERN
UNIVERSITY

Lower Level Optimization Communication

Optimization Services Description Language (OSDL)

```
<binding name="solverSoapBinding" type="impl:solver">
    <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="set">
        <wsdlsoap:operation soapAction=""/>
        <input name="setRequest">
            <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="encoded"/>
        </input>
        <output name="setResponse">
            <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="encoded"/>
        </output>
    </operation>
    <operation name="solver">
        <wsdlsoap:operation soapAction=""/>
        <input name="solverRequest">
            <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="encoded"/>
        </input>
        <output name="solverResponse">
            <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="encoded"/>
        </output>
    </operation>
    <operation name="solve">
        <wsdlsoap:operation soapAction=""/>
        <input name="solveRequest">
            <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="encoded"/>
        </input>
        <output name="solveResponse">
            <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="encoded"/>
        </output>
    </operation>
</binding>
<service name="solverService">
    <port binding="solverSoapBinding" name="solver">
        <wsdlsoap:address location="http://localhost:8080/axis/solver.jws"/>
    </port>
</service>
</OSDL:definitions>
```

Lower Level Optimization Communication

Optimization Services Flow Language (OSFL)

- Reserved for now
- Exact purpose is clear and may be necessary. But don't know of an appropriate design
- Intention is to organize analyzers, solvers, optimization simulations and other Optimization Services components, orchestrate information (e.g. input and output), sequence optimization process, resolve common variables etc.
- May prove to be especially useful in multi-objective, multi-start, multi-level, multi-disciplinary, Multi-processor optimization, Pareto-set optimization
- It is likely that OSFL will highly leverage on the interfaces specified in OSDL.
- May also need to collaborate with OSPL
- OSFL will probably wait to see the success and popularity of other OSXL.

Lower Level Optimization Communication

Optimization Services Endpoint Language (OSEL)

- Reserved for now
- Exact purpose not clear (may overlap with grid computing, thus unnecessary to design it)
- Intention is to be compatible with certain grid computing features
- Mainly to describe non-functional characteristics of an Optimization service
  - Quality of Service
  - Privacy policy
  - Auditing policy
- Should not affect the core syntax of OSDL
- May affect whether the solver requestor chooses to collaborate with a particular solver provider
- Can be important for asynchronous message flows (that is not request and response model)
  - Expected optimization time
  - Possible duration estimates for interaction or number of acceptable retires.
  - Basis on which solver requestor could establish time-out behavior, execute rollback or other interaction/compensation mechanism
  - Certain other run time information (may need to collaborate with OSPL)

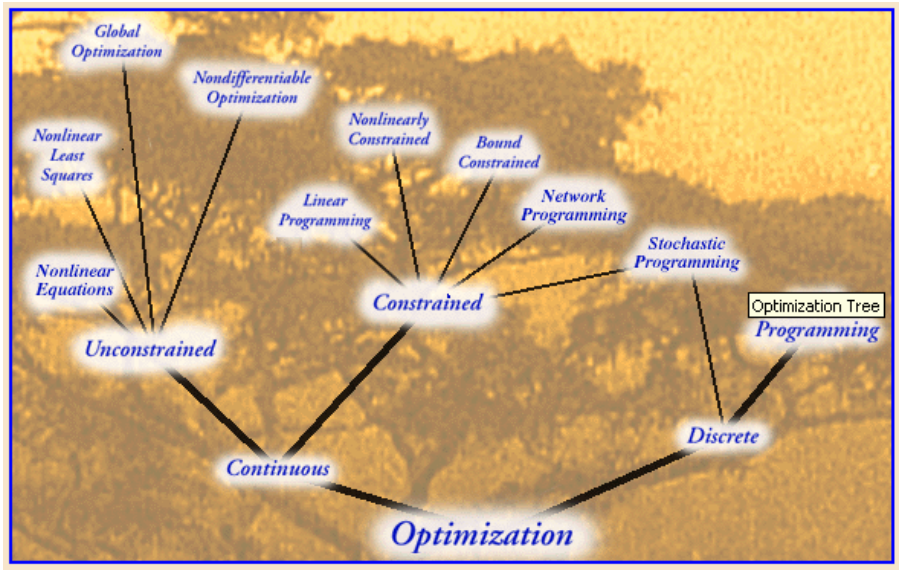
Outline

- Abstract
- Background
  - A General Picture – The Future of Computing
  - Our Positioning – The Hierarchy of Operations Research
  - Some Terminologies (Parallel/Distributed/Grid Computing, XML, Web Services, etc.)
- Motivation
  - Motorola's VP Intelligent Optimization System
  - AMPL-NEOS
- The Design and Framework for Distributed Optimization (NSF Funded)
  - Proposing and Defining Optimization Services (OS, OSXL)
  - Design of Distributed Optimization Architecture (Centralized and Decentralized)
  - Optimization Services Representation (OSTL, OSRL, OSOL, OSSL, OSAL)
  - Optimization Services Communication (OSCL, OSDL, OSFL, OSEL)
  - Optimization Services Inspection and Discovery (OSIL, OSPL, OSBL, OSQL)
- Future Work and Conclusion

Lower Level Optimization Communication

Optimization Services Inspection Language (OSIL)

- Think of optimization tree
- Think of it as record in database, only that the record is in XML rather than a row, and likely to be queries by OSQL rather than SQL.
- Describing keywords, abstract, description, etc.
- Describes Functionalities
  - OSOL
  - OSBL
  - Function type/format accepted
  - NEOS ranking ☺
- May also provide links to other solvers
- This is the piece that probably most need the authorities:
  - INFORMS
  - OTC/NEOS
  - W3C



Lower Level Optimization Communication

Optimization Services Inspection Language (OSIL)

```
<OSIL:inspection xmlns="http://schemas.xmlsoap.org/ws/2001/10/inspection/">
    <!--siimlar to a record in a table-->
    <!--can use OSQL to search the registry table of OSIL-->
    <abstract/>
    <service>
        <name>CPLEX</name>
        <abstract>A solver that solves linear problem</abstract>
        <description referencedNamespace="http://schemas.xmlsoap.org/wsdl/" location="http://localhost:8080/cplex?wsdl">
    </description>
        <solverCategory>Linear Programming</solverCategory>
        <OSOL>
            <standard/>
            <specific/>
        </OSOL>
        <OSBL/>
        <FunctionTypesAccepted>
            <MathML/>
            <binary>
                <languages>
                    <language>java</language>
                    <language>c++</language>
                </languages>
                <platforms>
                    <platform>win2000</platform>
                    <platform>UNIX</platform>
                </platforms>
            </binary>
            <webService/>
        </FunctionTypesAccepted>
        <NEOSRank>1 </NEOSRank>
    </service>
    <link referencedNamespace="http://schemas.xmlsoap.org/ws/2001/10/inspection/" location="http://localhost:8080/othersolver.wsil">
        <abstract>Other solver</abstract>
    </link>
</OSIL:inspection>
```

Lower Level Optimization Communication

Optimization Services Process Language (OSPL)

- Runtime description of solvers
  - Status
  - Number of jobs in its queue
  - Etc.
- Contains 2 sections
  - Standard
    - Common names of solver process information
    - Contains optional description (because it is standardized)
    - Can be used for solver inspection and discovery
  - Specific
    - Solver specific
    - Better contains a description for each option
    - Process values are typeless

Lower Level Optimization Communication

Optimization Services Process Language (OSPL)

```
<OSPL>
    <standard>
        <P name="status">
            <description>indicating the status of a solver server</description>
            busy
        </P>
    </standard>
    <specific>
        <P name="serverS1">
            <description>...</description>
            abc
        </P>
    </specific>
</OSPL>
```

Lower Level Optimization Communication

Optimization Services Benchmark Language (OSBL)

- Establishing industry standard for comparing solver algorithms and performances [w/ H. Mittelmann, U of Arizona]
- Benchmarking Solver at NEOS
- Probably doesn't need to contain benchmarker specific information, since benchmarking is supposed to be carried out against one single authoritative benchmarker.
- Not sure about exact design – will be designed by researchers who do benchmarking analysis.
- Not sure where it fits and how it can incorporated in the entire framework (Maybe its separate)
- Purpose is to make solver benchmark runs and make comparable results
- Can be coupled with OSIL discovery and inspection

Lower Level Optimization Communication

Optimization Services Query Language (OSQL)

- A language specification on query syntax
- To facilitate discovery of optimization solvers
- May just leverage on XMLQuery

Outline

- Abstract
- Background
  - A General Picture – The Future of Computing
  - Our Positioning – The Hierarchy of Operations Research
  - Some Terminologies (Parallel/Distributed/Grid Computing, XML, Web Services, etc.)
- Motivation
  - Motorola's VP Intelligent Optimization System
  - AMPL-NEOS
- The Design and Framework for Distributed Optimization (NSF Funded)
  - Proposing and Defining Optimization Services (OS, OSXL)
  - Design of Distributed Optimization Architecture (Centralized and Decentralized)
  - Optimization Services Representation (OSTL, OSRL, OSOL, OSSL, OSAL)
  - Optimization Services Communication (OSCL, OSDL, OSFL, OSEL)
  - Optimization Services Inspection and Discovery (OSIL, OSPL, OSBL, OSQL)
- Future Work and Conclusion

Applications

- Virtual Prototyping (T. Tirpak)
- iNEOS (J. Nocedal)
- IDEAL (W. Chen)
  - Experimental Design, Response Surface Methodology, Metamodeling
  - Multidisciplinary Optimization
  - Multi-level Optimization
- Stochastic Programming – Decomposition and Parallel Processing
- Integer Programming – Branch and Bound
- Optimization Via Simulation
- Data Mining/Machine Learning

Possible Future Process
The W3C Model

"I wrote in 1990 a program called "WorlDwidEweb"…

The first three years were a phase of persuasion to get the Web adopted …

In 1992 academia, and in 1993 industry, was taking notice …

After much discussion I decided to form the World Wide Web Consortium in September 1994, with a base at MIT in the USA …"

T. Berners-Lee

The W3C Process Model:

1. Working Group Notes
2. Working Drafts
3. Candidate Recommendations
4. Proposed Edited Recommendations
5. Proposed Recommendations
6. Recommendations

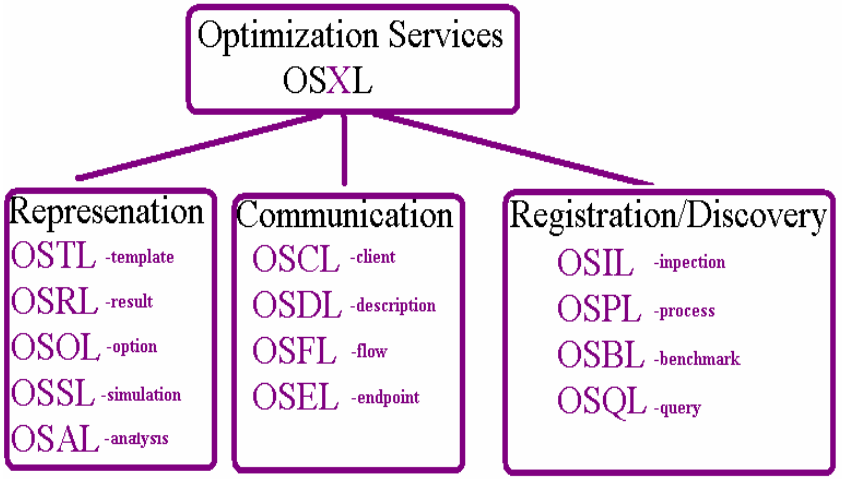
Future Work

- To be
  - Smoother
  - Simpler
  - More scalable
  - More systematic
- Elaboration
- Collaboration
- Standardization
- Finalization
- To serve as a basis for other research

NORTHWESTERN UNIVERSITY

Conclusion

- Distributed Optimization Design
  - Centralized
  - Decentralized
- Optimization Services
  - Representation (OSTL, OSRL, OSOL, OSSL, OSAL)
  - Communication and Scheduling (OSCL, OSDL, OSFL, OSEL)
  - Analyzing, Benchmarking and Categorizing (OSIL, OSPL, OSBL, OSQL)



OSXL

A	B	C	D	E	F	G
Analysis	Benchmark	Client	Description	Endpoint	Flow	
H	I	J	K	L	M	N
	Inspection				(Not to be used)	
O	P	Q	R	S	T	
Option	Process	Query	Result	Simulation	Template	
U	V	W	X	Y	Z	
			(*)			

NORTHWESTERN UNIVERSITY

Acknowledgement

- To Professor Fourer – for providing the vision and direction.
- To Tom Tirpak – for providing the perfect environment, opportunity and motivation
- To Professor John Birge and Professor Wei Chen – for serving on my committee
- To My wife, Haiyan Xu, who is graduating this afternoon.
- To all other OTC members for bring NEOS into being

Misc. Notes (ignore)

- Tag, name description value (option), parameter (attribute or element)
- Id and href and Soap message with attachment
- Web service used locally (conforming to ossl)
- Xml with binary attachment <pic xsi:type="soap-end:base64">*&^KJfak</pic>
- Use of ref or URI
- Constraint programming
- Complementarity problem
- Primitive type, array/matrix
- <obj> term and relation And closed form expression, name, can be web service choices <web service function> param, variable<>
- Data provider and model provider
- anyURI datatype usually a url
- URIReference: abs or relative
- <SOAP-ENC:uriReference>**http://www.zvon.org**</SOAP-ENC:uriReference>
  <foo xsi:type="SOAP-ENC:uriReference">**http://www.zvon.org**</foo> <SOAP-ENC:uriReference>**reference.xml**</SOAP-ENC:uriReference>
  <foo xsi:type="SOAP-ENC:uriReference">**reference.xml**</foo> <SOAP-ENC:uriReference>**schema.xml#uriReference**</SOAP-ENC:uriReference>
  <foo xsi:type="SOAP-ENC:uriReference">**schema.xml#uriReference**</foo>

NORTHWESTERN UNIVERSITY

Using XSL