Program Advisory Committee:

California Energy Commission Compliance Software

Agenda:

• Project Overview	(9:00am – 9:30am)
• Purpose of PAC	(9:30am – 9:45am)
• Compliance Software Overview	(9:45am – 10:15am)
• Data Model Overview	(10:15am – 10:45am)
• Break	(10:45am – 11:00am)
• Software Functional Requirements	(11:00am – 11:30pm)
• Ruleset Overview	(11:30am – 12:00pm)
• Lunch	(12:00pm – 1:00pm)
• Simulation Engine Interaction	(1:00pm – 1:30pm)
• User Interface Requirements	(1:30pm – 1:50pm)
• Compliance Forms	(1:50pm – 2:00pm)
• Pilot Projects	(2:00pm – 2:30pm)
• ACM Reference Method	(2:30pm – 2:50pm)
• Wrap-up	(2:50pm – 3:00pm)

Project Overview

- Warren-Alquist Act Requirements
- Technical Support Contract Details
- Project Team
- Schedule

Project Overview

- Background: The Warren-Alquist Act
  - Requirement for a "***public domain*** computer program…to estimate the energy consumed by buildings" in order to implement the Title 24 Standards.
- How this requirement was implemented in past code development cycles
  - Reliance on proprietary software

Project Overview

- **Technical Support Contract Details**
  - Contract #400-10-015 – Nonresidential Building Energy Science Technical Support
  - *Begin the process of **collaboratively** developing, testing, documenting, and supporting **open source**…building energy analysis tools used for Standards development, Standards compliance and other energy efficiency and clean energy public policy implementation.*

Project Overview

- Focus on **Compliance**
  - Software will indicate if a building is compliant with the energy code
  - Incorporation of the energy code ruleset will be primary focus
  - We are not developing simulation algorithms
  - We are not competing with commercial vendors

Project Overview

- Why Open Source Compliance Software?
  - Meets requirements of Warren-Alquist Act
  - Encourages collaboration amongst codes and standards organizations
  - Enables maintenance and updates
  - Encourages innovation by 3rd parties without sacrificing standardization

Open Source Licensing Requirements

- open use by both free and commercial products w/ no requirement for submitting changes back to the repository
- Minimal restrictions on use, and distribution

Project Overview

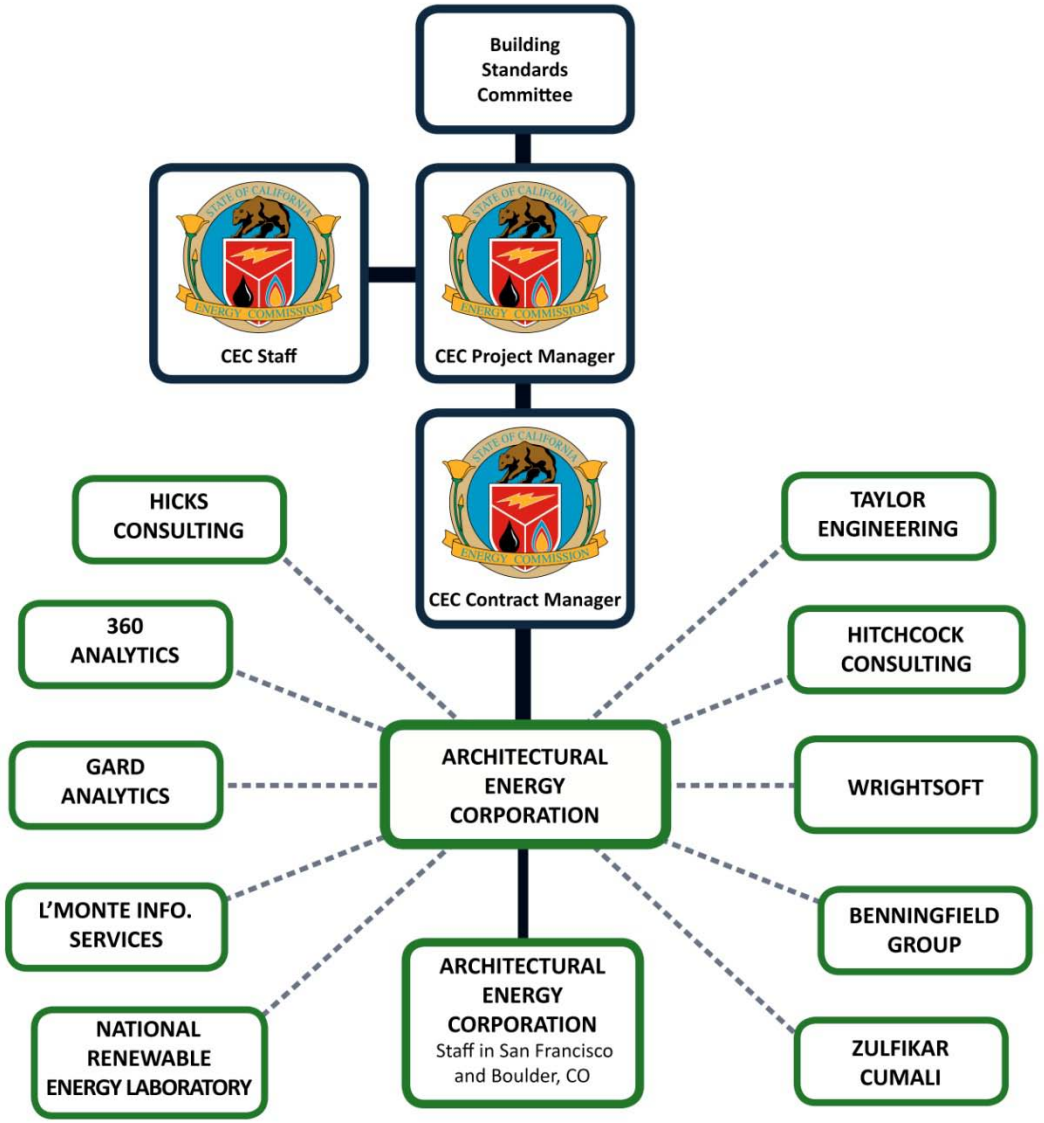
- EnergyPlus as Reference Software
  - Publicly funded software
  - Refer to LBNL report assessing capabilities of E+ for Title 24 Compliance
  - Ongoing E+ development efforts to address necessary modeling features

Project Overview

- Compliance Paths
  - Utilize the CEC public domain Compliance Software
  - Utilize commercial software that has implemented the rules processing features (Compliance Engine) but has an alternate user interface and/or simulation engine
  - Utilize approved software that has a custom implementation of the Title 24 performance rules

Project Overview

- Project Team
  - CEC
  - AEC
  - Subject Matter Experts



2013 Standards
Nonresidential ACM Schedule

Completion Date	ACM Component
March 2012	ACM Reference Method
March 2012	Draft ACM Reference Manual
April 2012	Compliance Software Proof Of Concept
October 2012	Final Compliance Engine
January 2013	Final ACM Reference Manual
February 2013	Compliance Engine Pilots
February 2013	Final Compliance Software
April 2013	Commission Certification of Compliance Software
April 2013	Commission Approval of ACM Reference Manual
July 2013	Deadline for Compliance Software Certification

Purpose of Program Advisory Committee (PAC)

- Provide guidance, input, review, and comment to the project team
- Review technical materials related to project approaches and raise any issues
- Provide recommendations on open issues
- Project team will document and respond to feedback from the PAC

PAC Members

- The PAC consists of industry experts in the fields of:
  - energy simulation,
  - mechanical system design and operations,
  - members of the trades who will apply the results of the project,
  - software product developers,
  - professionals with open source software management experience,
  - California utility representatives with program responsibilities in areas requiring building energy simulation tools

PAC Members

- PAC member introductions

Compliance Software Overview

- Target Audience
- Use Case Scenarios
- Elements of the Software
- Documentation
- Quality Assurance
- Licensing Approaches

Compliance Software Overview

- Target Audience
  - Building Designers (“Users”)
    - hands-on users of the software to evaluate whether their buildings comply with the standards
  - Third-Party Software Vendors (“Vendors”)
    - may incorporate compliance analysis into their software tools and provide more robust feature set
  - Codes, Standards, and Incentives Programs (“Regulators”)
    - may adapt software to perform analysis for programs in addition to California Title 24

Compliance Software Overview

- **Use Case Scenario 1: Building Designers**
  - User specifies the PROPOSED building design in the software interface
  - Software will review the proposed building, and make any necessary changes for the compliance analysis of the proposed building
  - Software will automatically create a BASELINE building (representing a minimally code compliant building) according to rules defined by mandatory, prescriptive, and performance Standards (Title 24, Part 6 and the Alternative Calculation Method ("ACM"))

Compliance Software Overview

- **Use Case Scenario 1: Building Designers (cont'd)**
  - Software will send the PROPOSED and BASELINE building models to an external energy modeling simulation engine
  - Simulation engine will perform equipment sizing and annual energy simulations
  - Output results from energy simulations will be passed back to the compliance software and software will assess if the building is compliant with the Standards
  - Software will generate Compliance forms and reports

Compliance Software Overview

- **Use Case Scenario 2: Vendors**
  - 3rd Party Software Vendors may incorporate the compliance analysis and reporting features of the software into their proprietary tools by taking advantage of data exchange protocols of the compliance software
  - User Interface may be replaced by more robust Graphical User Interfaces
  - Simulation Engine may be replaced by alternate engines
  - Compliance Analysis features would function in the same manner as previously described (3rd Party Vendors would NOT control compliance rules)

Compliance Software Overview

- **Use Case Scenario 3: Regulators**
  - Software Ruleset may be adapted for use with alternate energy codes, standards, beyond-code programs, or incentive programs
  - Software functionality would remain the same as previously described, except the BASELINE building would be created to represent an alternate code
  - Regulators are assured that the ruleset is properly interpreted since they are responsible for its creation
  - May also be used for analysis during the code development cycle for impact assessments

Compliance Software Overview

- **Elements of the Software**
  - Graphical User Interface
  - Data Model
  - Ruleset files
  - The Compliance Engine
  - Interaction with Simulation Engine
    - OpenStudio & EnergyPlus
  - Report Generator

  - Software Diagram

Compliance Software Overview

- **Documentation**
  - Functional Requirements
  - Specifications
  - Quality Assurance Plan
  - Development Report
  - Licensing Report

Compliance Software Overview

- Questions?

Data Model Overview

- “High-level” model for compliance analysis vs. “Detailed” BEM model
- Overview of Standards Data Dictionary
- SDD XML
- Mapping to existing data models (gbXML, IFC)

“Standards Data Model” vs. “Building Energy Model”

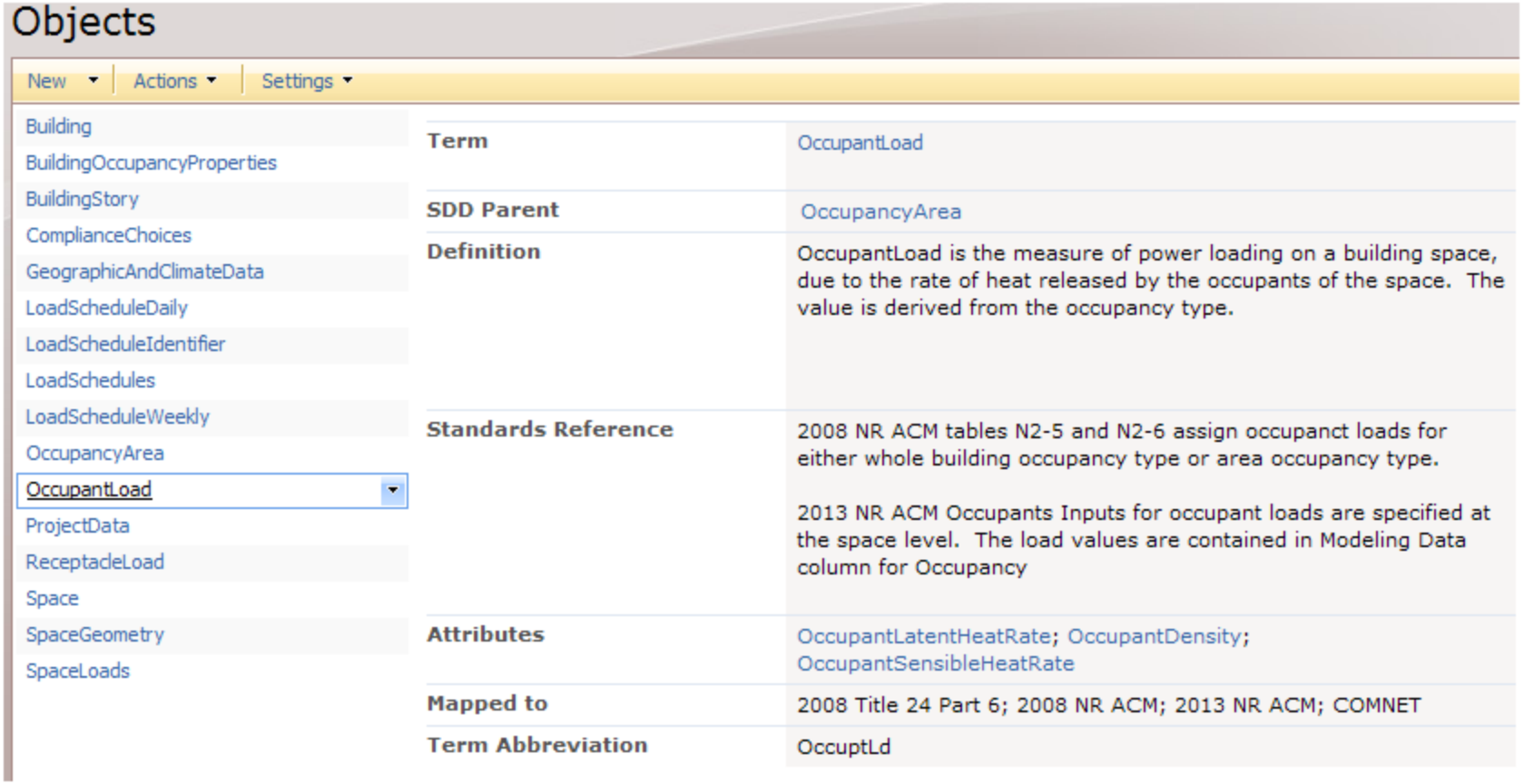
- Standards Data Model
  - Building parameters needed to represent and trigger compliance rules
  - Parameters that are set by these rules
- Building Energy Model
  - Energy simulation engine I/O parameters
- There is no existing publicly-available schema that has been developed from the Standards View perspective.

Title-24 Standards Data Dictionary (SDD)

- SDD terminology is basis for internal native data model within the ACM Standards Compliance Engine Software
- To facilitate communication between the Compliance Engine and external modules, an XML schema (XSD) based on the SDD is also being developed
- To the extent possible, the SDD XSD will be consistent with existing BEMs

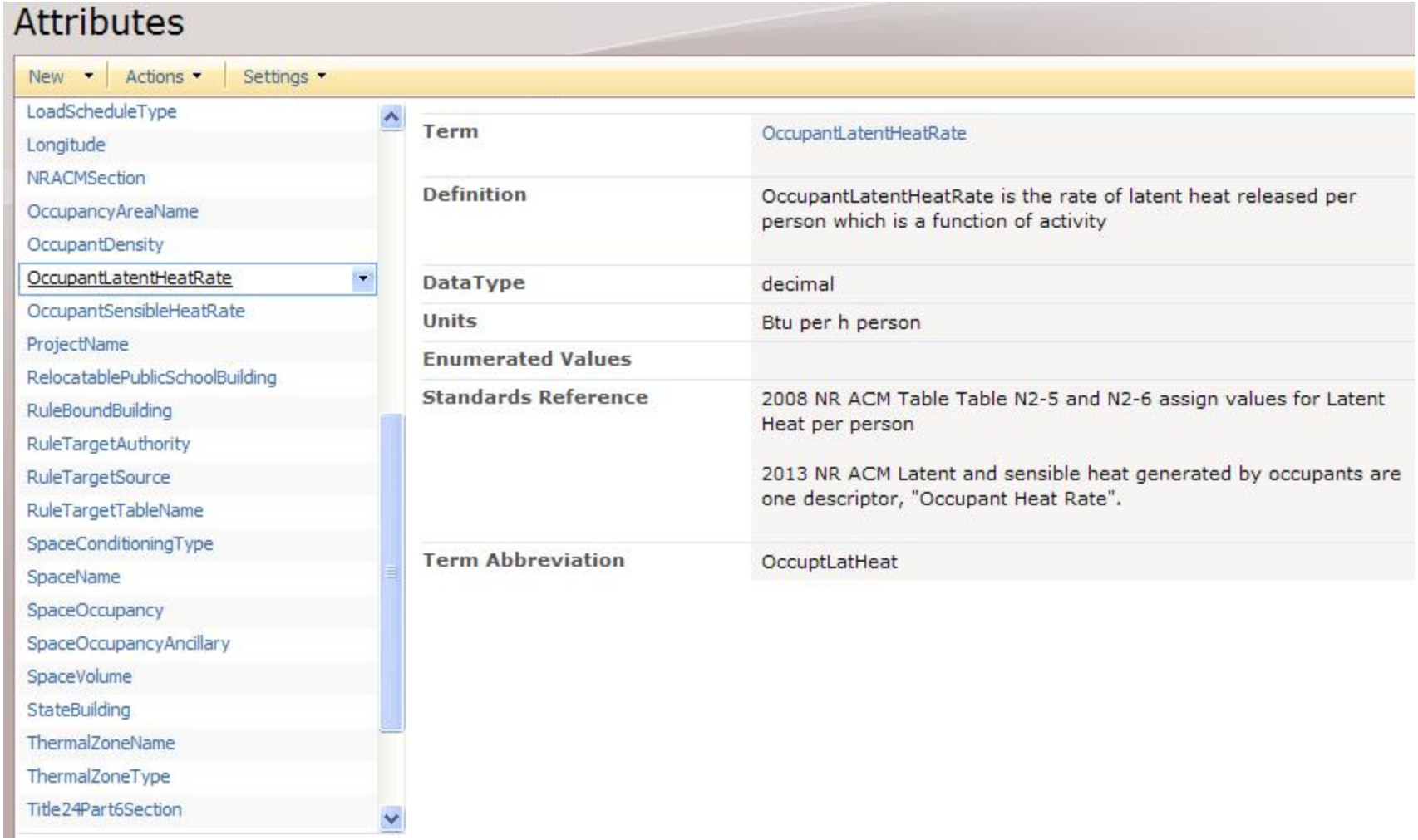
Data Model Overview: Standards Data Dictionary (SDD)

- What is in the Standards Data Dictionary?
  - Terms from the building energy compliance domain used in compliance software.
  - Information includes term name, data type, meaning, units, related terms
  - Naming conventions for capitalization, abbreviations and acronyms
- Example screen shots of SDD lists:



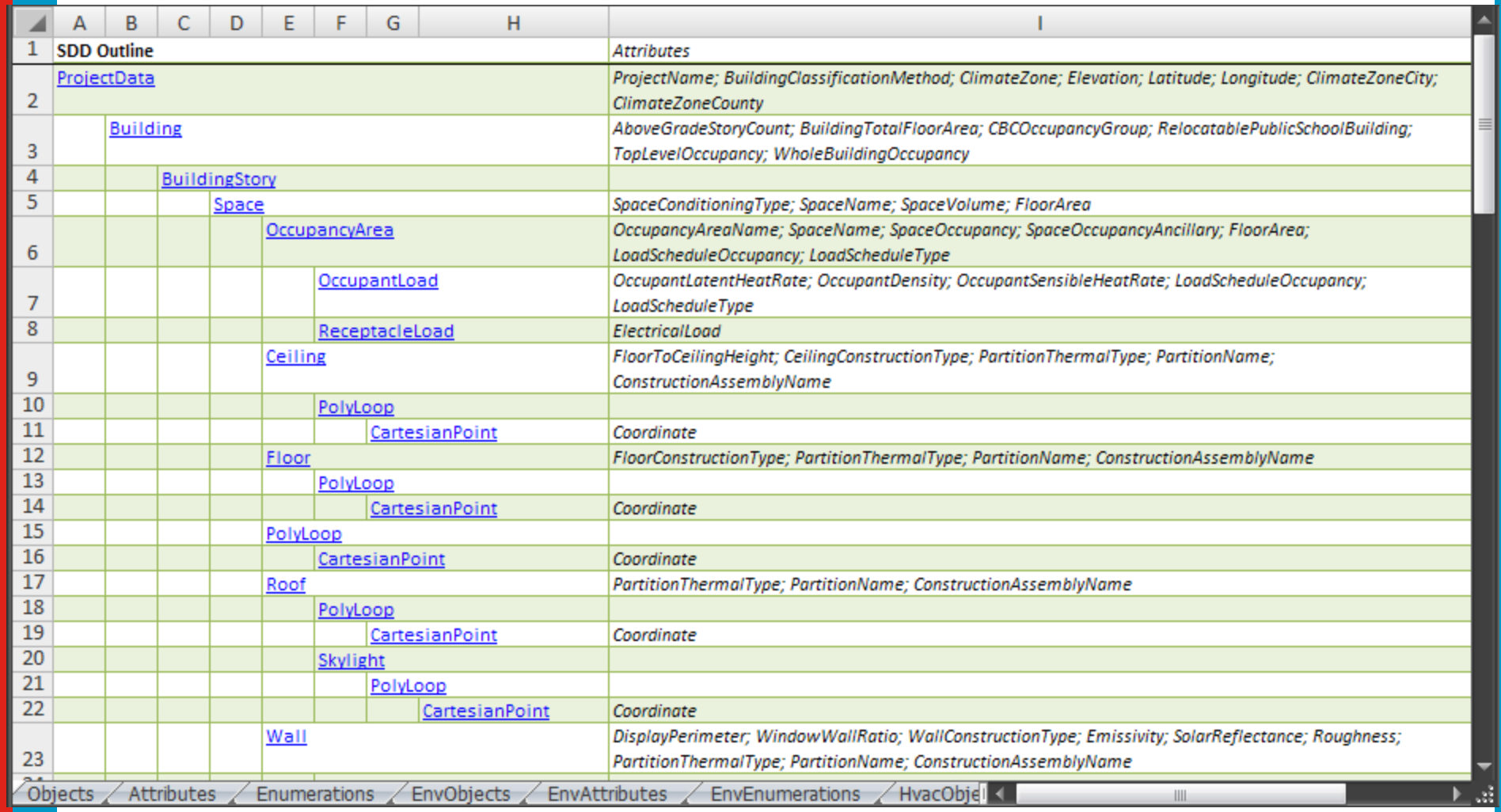
Data Model Overview: Standards Data Dictionary

- Attribute of the OccupantLoad



Data Model Overview: Standards Data Dictionary

- SDD can be exported to an Excel Workbook.



	A	B	C	D	E	F	G	H	I
1	**SDD Outline**								*Attributes*
2	ProjectData								*ProjectName; BuildingClassificationMethod; ClimateZone; Elevation; Latitude; Longitude; ClimateZoneCity; ClimateZoneCounty*
3		Building							*AboveGradeStoryCount; BuildingTotalFloorArea; CBCOccupancyGroup; RelocatablePublicSchoolBuilding; TopLevelOccupancy; WholeBuildingOccupancy*
4			BuildingStory						
5				Space					*SpaceConditioningType; SpaceName; SpaceVolume; FloorArea*
6					OccupancyArea				*OccupancyAreaName; SpaceName; SpaceOccupancy; SpaceOccupancyAncillary; FloorArea; LoadScheduleOccupancy; LoadScheduleType*
7						OccupantLoad			*OccupantLatentHeatRate; OccupantDensity; OccupantSensibleHeatRate; LoadScheduleOccupancy; LoadScheduleType*
8						ReceptacleLoad			*ElectricalLoad*
9					Ceiling				*FloorToCeilingHeight; CeilingConstructionType; PartitionThermalType; PartitionName; ConstructionAssemblyName*
10						PolyLoop			
11							CartesianPoint		*Coordinate*
12					Floor				*FloorConstructionType; PartitionThermalType; PartitionName; ConstructionAssemblyName*
13						PolyLoop			
14							CartesianPoint		*Coordinate*
15					PolyLoop				
16						CartesianPoint			*Coordinate*
17					Roof				*PartitionThermalType; PartitionName; ConstructionAssemblyName*
18						PolyLoop			
19							CartesianPoint		*Coordinate*
20						Skylight			
21							PolyLoop		
22								CartesianPoint	*Coordinate*
23					Wall				*DisplayPerimeter; WindowWallRatio; WallConstructionType; Emissivity; SolarReflectance; Roughness; PartitionThermalType; PartitionName; ConstructionAssemblyName*

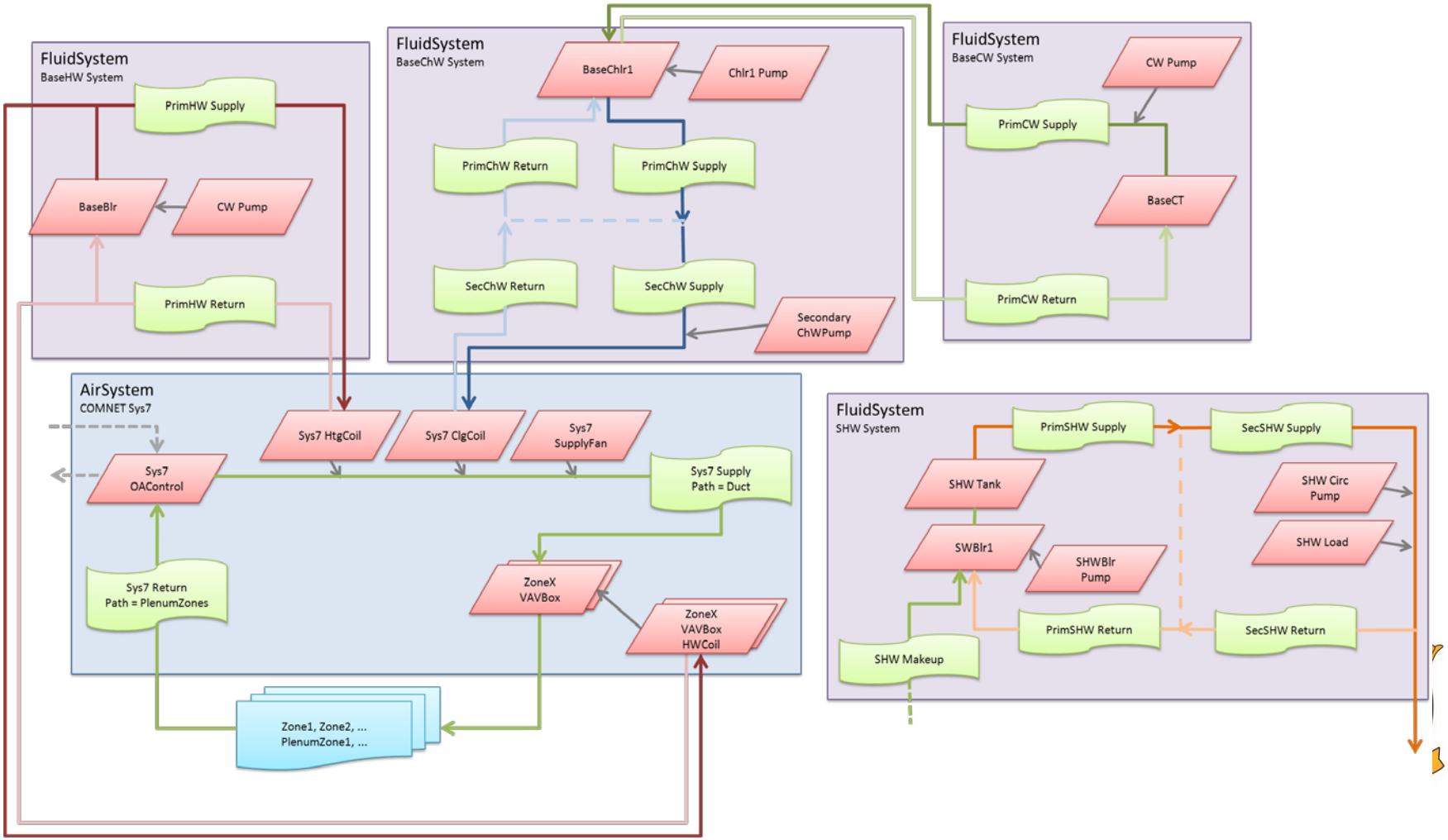
Objects / Attributes / Enumerations / EnvObjects / EnvAttributes / EnvEnumerations / HvacObje

Data Model Overview: HVAC

- **HVAC Data: Component Based System Representation**
  - Thermal Zones are served by conditioning systems
    - Air Systems and Fluid Systems
      - Sub-Components include fans, coils, terminal units, chillers, boilers, etc.
  - Provides the flexibility to describe both conventional and more innovative HVAC systems
  - Component data is sufficient to capture elements in the standards and to be adequately interpreted by BEM
    - But not tied to the data structure of a specific simulation engine

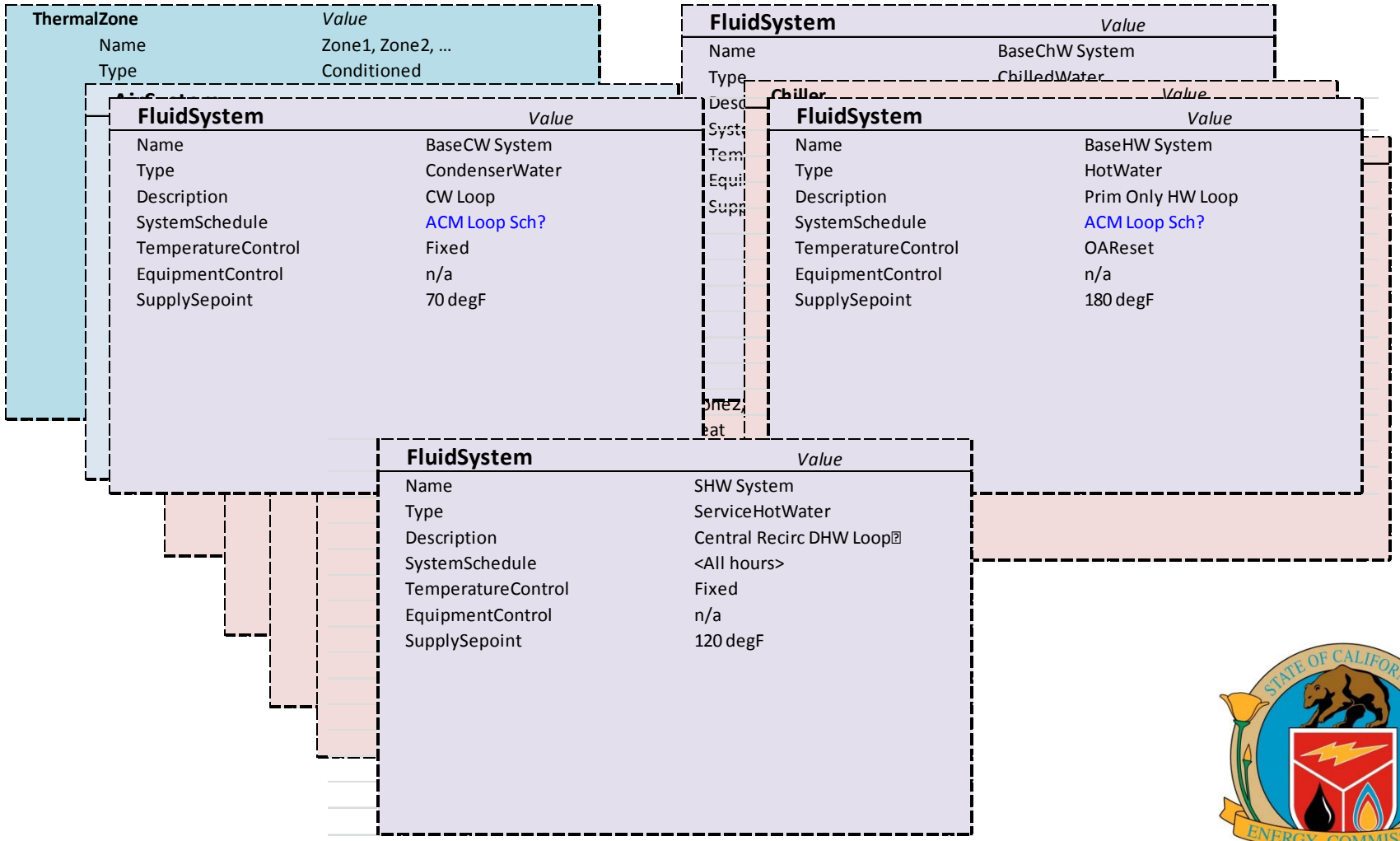
Data Model Overview: HVAC

- Example Representation of a Conventional HVAC System:
  - VAV w/ Reheat; Cooling from Chilled Water Plant



Data Model Overview: HVAC

- Example Representation of a Conventional HVAC System:
  - VAV w/ Reheat; Cooling from Chilled Water Plant



ThermalZone	Value
Name	Zone1, Zone2, …
Type	Conditioned

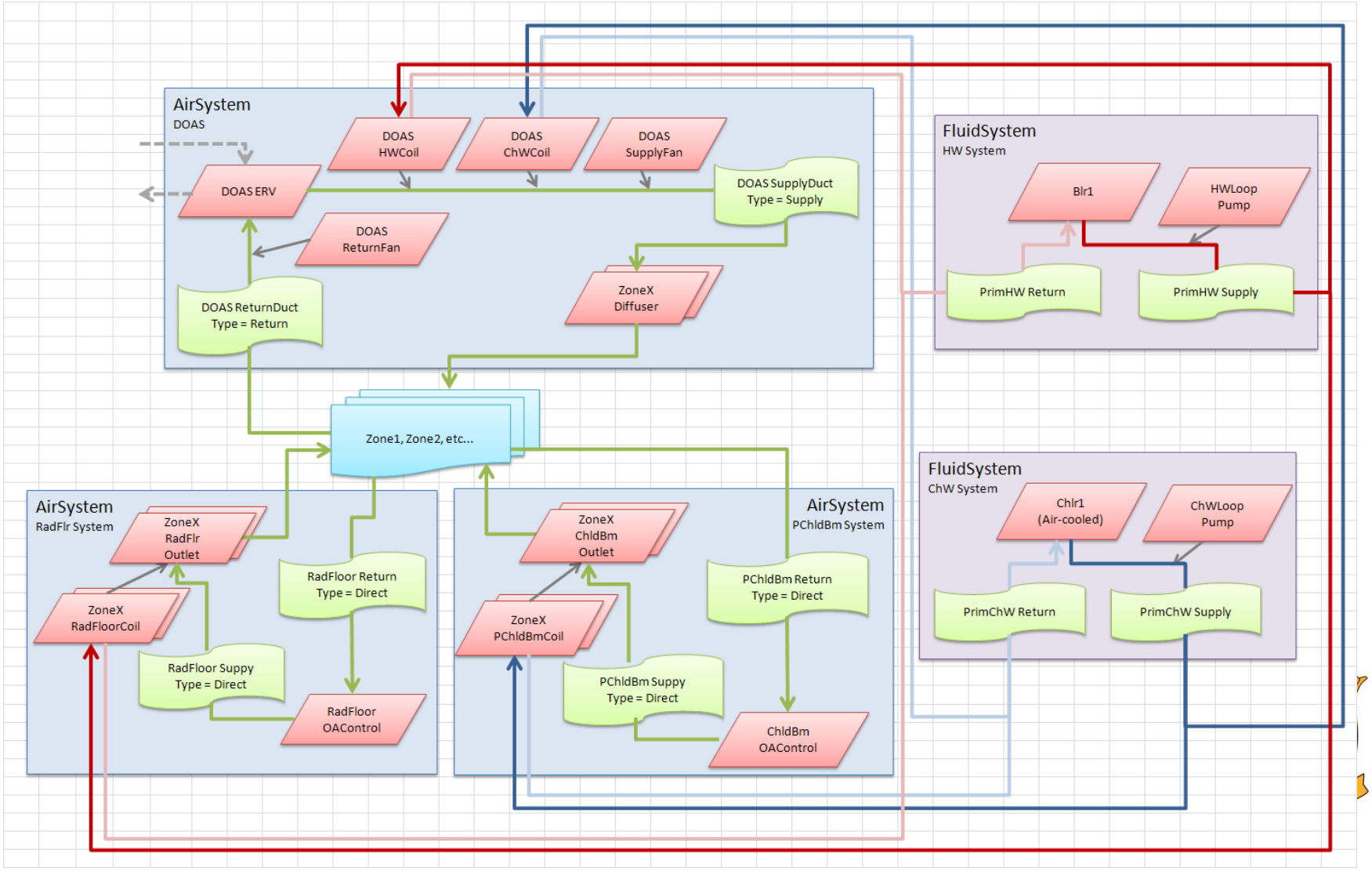
FluidSystem	Value
Name	BaseCW System
Type	CondenserWater
Description	CW Loop
SystemSchedule	ACM Loop Sch?
TemperatureControl	Fixed
EquipmentControl	n/a
SupplySepoint	70 degF

FluidSystem	Value
Name	BaseHW System
Type	HotWater
Description	Prim Only HW Loop
SystemSchedule	ACM Loop Sch?
TemperatureControl	OAReset
EquipmentControl	n/a
SupplySepoint	180 degF

FluidSystem	Value
Name	SHW System
Type	ServiceHotWater
Description	Central Recirc DHW Loop
SystemSchedule	<All hours>
TemperatureControl	Fixed
EquipmentControl	n/a
SupplySepoint	120 degF

Data Model Overview: HVAC

- Example Representation of a Advanced HVAC System:
  - DOAS w/ Radiant Floor Heating and Chilled Beam Clg



Data Model Overview: Standards Data Dictionary

- Why do we need the Standards Data Dictionary?
  - Improves data Integrity, the trustworthiness of the data throughout its processing
    - By creating a cohesive, consistent representation of the data used throughout the application
  - Improves interpretation of the data to give repeatable results & avoid defects from
    - Duplicate objects with different names
    - Irregular use of abbreviations or naming conventions
    - Irregular use of units
  - Provides documentation for developers to avoid these kinds of problems -
    - Mars Climate Orbiter was lost because one development team used imperial units for a measurement and another used metric and when the data was passed between the two modules it wasn't converted to the other system and there went $125 million dollars (more, as this was in 1999)
    - A simple example closer to home (and less dramatic) defining a range – if you have a range between 1 and 20, do you mean inclusive of 1 and 20 or exclusive of both or … Imprecision caused problems with overlapping ranges and multiple options for assigning values
- Who benefits from these improvements
  - Compliance Engine software & our data partners

Data Model Overview: Standards Data Dictionary

- How is the SDD being used?
  - The SDD is the data foundation for the 2013 NR Compliance Engine
    - Rule set
    - External data exchange files (xml/xsd)
    - Internal data model
    - Compliance forms
  - The SDD also provides a shared controlled vocabulary for use in
    - HERS registries compliance document registration
    - CEC Compliance document repository
    - Building Energy Standards Modeler research program (BESM)
    - Other applications: Utility Incentive Programs, Other Codes and Standards such as ASHRAE 90.1

Data Model Overview: SDD XML/XSD

- XML & XSD: What is the difference?
  - XML stands for Extensible Markup Language
    - XML is a standard developed by the World Wide Web Consortium (W3C) for representing text based data
    - XML defines how to use tags to mark the beginning and end of an individual piece of data and how to relate different data by nesting data in hierarchical relationships.
    - There are no rules on what data is called, limits on the data or how it is organized.
    - The file extension is ".xml".
  - XSDL stands for XML Schema Definition Language
    - XSDL is also a standard developed by the W3C which defines mechanisms for representing rules about the data in an XML file, in order to standardize the data exchange.
    - This file has an extension of "xsd"
  - The XSD
    - Implements the rules and constraints defined for data in the Standards Data Dictionary
    - Applies them to the XML instance file during validation

Data Model Overview: SDD XML/XSD

- SDD XSD
  - The SDD XSD file is the contract between the compliance engine data exchange partners defining what is required and allowed in the xml file
    - What data can be represented, what it is called, whether it is required or optional, what are acceptable values for the data, and how data is organized.
  - Start with the SDD terms, definitions and structure
  - Define data validation to embed in the XSD schema file for the XML file
    - Essential for data partners to use as a specification for developing their translator, allowing them to validate their SDD.xml – instance files against the SDD.xsd schema file
    - Examples:
      - A text field has a limited number of acceptable values - define the type as enumeration and list of values in the XSD
      - A numeric field only accepts values within a acceptable range – define upper/lower bounds.
      - Ensuring that references to other objects actually exist in the xml file by defining identity constraints

SDD XSD Design Criteria

- The SDD XSD must support SDD terms for representing and processing rules
  - It must be consistent format for both internal processing and data exchange between modules
  - A single representation of any given element, not multiple possibilities
    - Example: Geometry representations

- Insufficient resources to implement multiple format import transformations
- Define where to draw the line between
  - Simple representation for rules processing but enough detail for simulation
  - Representing everything in the building model or passing through elements or systems not touched by rule processing but required for simulation.
- When possible adopt better terms and data organization from existing schemas – Issues raised from this objective include
  - Schemas are uneven – some strong in one area and weak in another, which would lead to taking bits from different schemas
  - Mash-up of different existing schema data type systems and naming conventions would compromise the consistent data model expressed in the SDD, and require 3rd party vendors to be aware of differences reducing anticipated ease of implementation.
  - Some schemas reviewed are still under development and that means working with a moving target that has no history of use, adoption, how frequently it changes or how versioning will be handled
  - This led to mapping SDD terms to existing data models…

Mapping to Existing Data Models

- Leverage existing models
  - Not reinvent wheels that actually turn
  - Facilitate SDD XML instance file creation by third-party tools
  - Facilitate data exchange
- Cognizant of related existing and developing data models
  - gbXML, IFC, simModel, OpenStudio Model, EnergyPlus IDD
- Established relationships with related ongoing projects
  - OpenStudio, Simergy, DOE Interoperability Project, BESM, COMNET
- Involved in ongoing mapping efforts between models
- Incorporating existing structure and elements where applicable
  - IFC, OSM, simModel similar Project.Building.BuildingStory.Space.Surface
  - gbXML detailed geometry elements

Data Model Overview

- Questions?

Break

- 10:45am – 11:00am

Software Functional Requirements

- Detailed review of software functionality

Software Functional Requirements

- Purpose:
  - Describe functionality of individual software system components
  - How system components work together
  - Ensure system meets project objectives
    - Timeline & budget
    - Open source
    - Shared non-residential/residential usage
    - Division of labor across components
    - Security of analysis and reports
    - Maintainability of software system

Scope & Organization

- Summary of entire software system
- Principal module coverage:
  - Compliance (rules) engine
  - EnergyPlus interface (via link to OpenStudio)
  - User interface
  - Compliance reporting module
- System performance
- Appendices

Compliance (rules) Engine

- Managing compliance rulesets
  - Text/CSV-based ruleset source -> binary
- Supervise analysis
  - Manage building models & rule evaluation
  - Communication w/ other system modules
    - Initiate multiple building energy simulations
    - Initiate generation of reports
    - Progress and messages available to UIs

EnergyPlus Interface (OpenStudio)

- Passing SDD building models to OS
- OpenStudio: SDD -> OSM -> E+
- Use of BCL (bldg comp library) objects
- Monitoring EnergyPlus simulation status
- Reporting translation and simulation messages
- Retrieval of simulation results

User Interface

- “Minimal” interface
- Access to all SDD data
- Native file format – SDD XML
- Supported platforms – Windows vs. others

Compliance Reporting

- Accessing analysis inputs and results
- Output format and security
- Report maintenance (by CEC staff)

Performance Requirements

- Performance by function/task:
  - Simulation speed (not addressed in project scope)
  - Rules processing
  - Building model translation
  - Report generation (addition)

Requirements Appendices

- Ruleset data & source format/approach
  - Data model approach (non-res)
  - Procedural approach (res)
- Rule syntax & expression reference
- Optional capabilities
  - Multi-threaded simulation support
  - Multi-platform UI

Software Functional Requirements

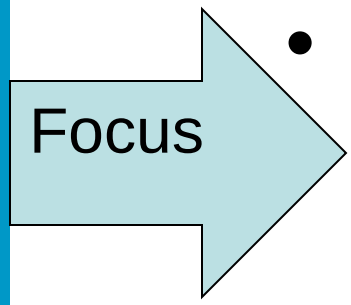
- Questions?

Ruleset Overview

- Review of Rule Syntax
- How rules are processed by compliance engine
- Maintaining integrity of the ruleset

Overview

- SDD XML – hierarchical building data representation
- Ruleset to transform user SDD XML into proposed SDD XML and baseline SDD XML
- Each SDD XML translated into energy simulation input and simulation results compared



Ruleset

- Operations / Purposes:
  - Define valid ranges, defaults, element descriptions & how rules apply the code
  - Validate user's proposed model
  - Apply proposed model restrictions
  - Transform proposed model into baseline

- Each rule file describes operation on major topic – opaque envelope, daylighting, ..

Ruleset Data

- Source files (maintained by rule authority)
  - Variety of tools enable rule creation/edit
  - More compatible w/ archiving/tracking repositories
  - Each rule file covers aspects of major topic (opaque envelope, daylighting, etc.)
- Distributed ruleset file
  - Individual source files compiled into single binary file (via CE) distributed with software tools
  - Security features of binary ruleset file combined with engine & reporting software modules prevent analysis manipulation
  - Compilation process includes some ruleset error checking

Principal Ruleset Data

- Model transformation definitions
  - Proposed (HVAC) Sizing / Proposed / Baseline Sizing / Baseline
  - Transformation name/abbreviation used in rules to identify SDD XML model elements for specific transformations
- Component Library
  - Construction assembly materials, schedules, etc.
- Lookup Tables
  - Loads by activity area, HVAC equipment efficiency limits, etc.

Sample Look-up Table

- Lookup tables can exist within rule text files or as exports from spreadsheets (as CSV files)
- Ruleset tables organized as they are defined in the code or ACM (as shown below)

TABLE143-A – PRESCRIPTIVE ENVELOPE CRITERIA FOR NONRESIDENTIAL BUILDINGS (INCL BUILDINGS WHERE MANUFACTURER CERTIFIES USE ONLY IN SPECIFIC CLIMATE ZONE; NO BUILDINGS AND GUEST ROOMS OF HOTEL/MOTEL BUILDINGS)

			Climate Zone								
			1	2	3	4	5	6	7	8	9
Roofs/Ceilings	Metal Building		0.065	0.065	0.065	0.065	0.065	0.065	0.065	0.065	0.065
	Wood Framed and Other		0.049	0.039	0.039	0.039	0.049	0.075	0.067	0.067	0.039
Roofing Products	Low-sloped	Aged Reflectance	NR	0.55	0.55	0.55	0.55	0.55	0.55	0.55	0.55
		Emittance	NR	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
Walls											

T24-Nonres-2013 Lookup Tables 11-12-13.xlsx

	A	B	C	D	E	F	G	H
1	;	2013 Title-24 Non-residential						
2	;	Table 143A - Prescriptive Envelope Criteria for Nonresidential ...						
3	Envelope143A	;		Climate Zones:				
4	SurfaceType	SurfaceDescrip	ReturnValue	CZ1	CZ2	CZ3	CZ4	CZ5
5	Roofs/Ceilings	Metal Building	U-factor	0.065	0.065	0.065	0.065	0.065
6	Roofs/Ceilings	Wood Framed and Other	U-factor	0.049	0.039	0.039	0.039	0.049
7	Roofing Products	Low-sloped	Aged Reflectance	NR	0.55	0.55	0.55	0.55
8	Roofing Products	Low-sloped	Emittance	NR	0.75	0.75	0.75	0.75
9	Roofing Products	Steep Sloped (less than 5 lb/ft²)	Aged Reflectance	NR	0.20	0.20	0.20	0.20
10	Roofing Products	Steep Sloped (less than 5 lb/ft²)	Emittance	NR	0.75	0.75	0.75	0.75
11	Roofing Products	Steep Sloped (5 lb/ft² or more)	Aged Reflectance	0.15	0.15	0.15	0.15	0.15
12	Roofing Products	Steep Sloped (5 lb/ft² or more)	Emittance	0.75	0.75	0.75	0.75	0.75
13	Walls	Metal Building	U-factor	0.113	0.061	0.061	0.061	0.061
14	Walls	Metal-framed	U-factor	0.098	0.062	0.082	0.062	0.062
15	Walls	Mass Light	U-factor	0.196	0.170	0.278	0.227	0.440

```
Envelope143A( u.proj.ClimateZoneAbbrev, SurfaceType="Roofs/Ceilings",
              SurfaceDescrip=u.bldg.story.roof.143ADescrip, ReturnValue="U factor" )
```

Individual Rule Data

- Data model ruleset approach consolidates all data by SDD element into rule “block”
  - Short & long element descriptions
  - Units and/or enumerated list (when applicable)
  - Soft & hard numeric range definitions
  - Default value (for user proposed design)
  - Settings for each ruleset transformation
- Rule Expressions
  - Default & settings portion of rule made up of expressions similar to spreadsheet formulas, w/ logical statements and functions designed to facilitate building model manipulation

Sample Rule

```
RULE building.construction.rValue
  DESCRIPTION         "Insulation R Value”
  HELP     "Multiple lines of help about the rules pertaining to this particular
           rule. The start and end double quotes can be on different lines"
  REFERENCE           NonResACM2.1
                      Section6.3Table23
  MINIMUM             0.001
  COMMONMINIMUM       8
  COMMONMAXIMUM       38
  MAXIMUM             100.00
  UNITS               h-ft2-F/Btu
  DEFAULT             11
  PROPOSEDSIZING
           u.building.construction.rValue
  PROPOSED
           u.building.construction.rValue
  BASELINESIZING
           IF building.climateZone > 3 THEN
               12
           ELSE
               14
           ENDIF
  BASELINE
           bz.building.construction.rValue
ENDRULE building.construction.rValue
```

Ruleset Overview

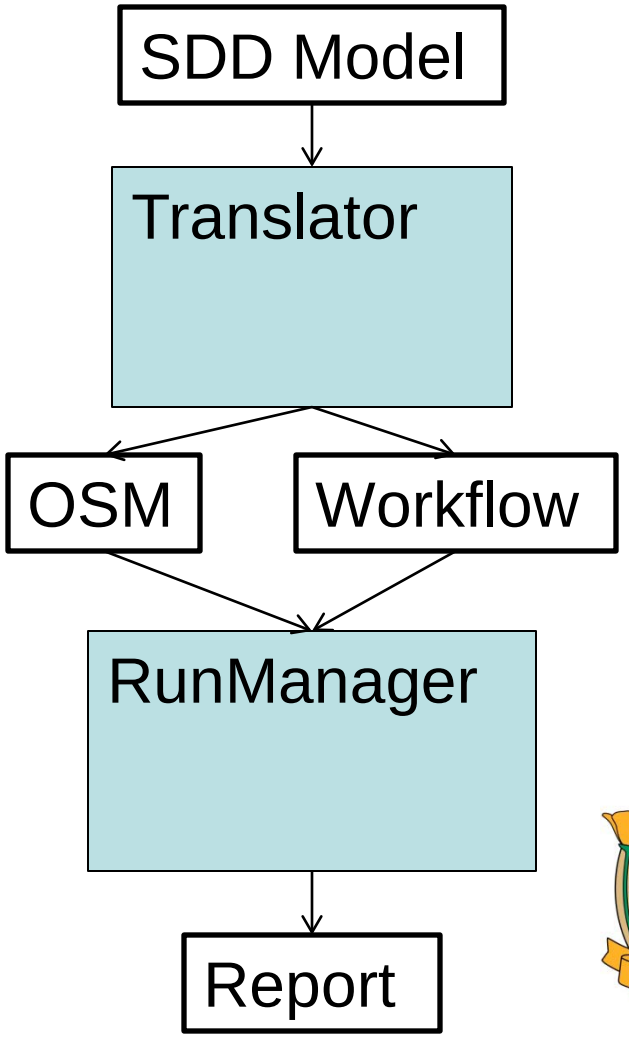
- Questions?

Lunch

- 12:00pm -1:00pm

Simulation Engine Interaction

- Data Translation
- Simulation Control
- Reporting



Simulation Engine Interaction

- OpenStudio Demo
  - Translation: EnergyPlus, gbXML, BCL
  - RunManager: Standalone, Embedded
  - Results: XML, ProjectDatabase, Logs, Reports

Simulation Engine Interaction

- Questions?

User Interface Requirements

- Ability to manage the SDD Model Data
  - Native File Format = SDD XML
  - Ability to read/write SDD XML
- Model Navigation
  - Hierarchical display of model components
- Ability to create new components, edit or delete existing components

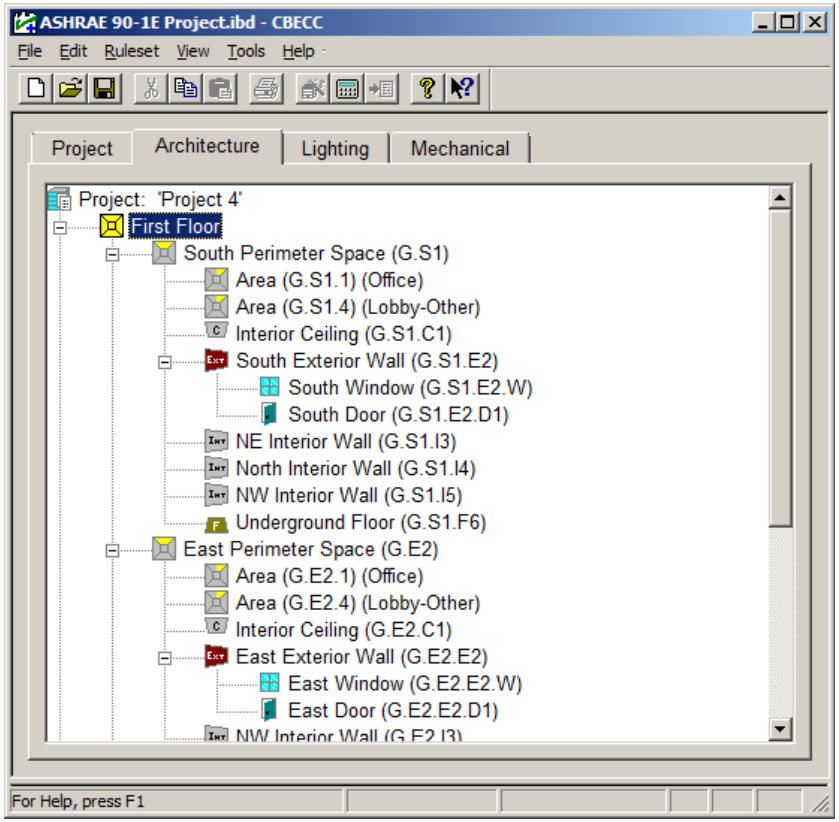
User Interface Requirements

- Provides visual indicators to track progress during rules processing or simulation processes
- Capability to view reports after completion of compliance analysis

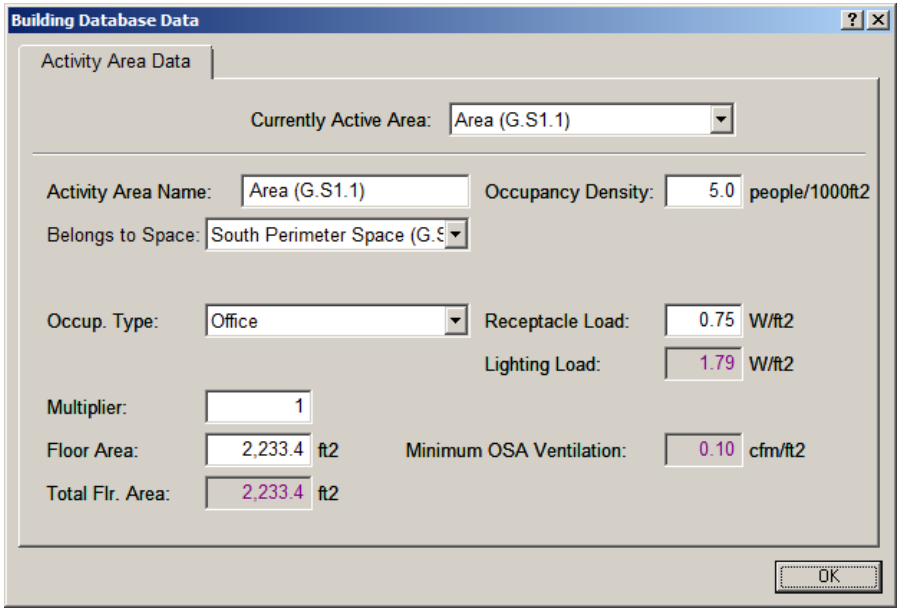
User Interface Requirements

- Will not include wizards, mouse-driven geometry creation, or highly-graphical building or HVAC system diagrams
- May include ability to display default values alongside user inputs

User Interface Requirements



Mock-up of Hierarchical Interface showing relationships of components



Mock-up of editor for individual components

User Interface Requirements

- Questions?

Compliance Forms

- Automated Forms Generation
- Software will automatically generate compliance forms

Compliance Forms Reporting Engine

- Using Compliance Forms Reporting Engine developed for the Residential ACM which uses these basic technologies :
  - XML Source Data Document - This is the format that all systems must be able to read and write.
  - XML Schema Document - This defines the structure and types of data for generating and validating conforming xml data documents.
  - XSLT Document - This is a means of transforming data in an xml data document into different forms for either display or converting data into different structures that use it. It will be the standard definition of how compliance data is transformed into a standard reporting structure
  - XSL-FO Document – This adds the Formatting Objects elements, which describe layout and formatting mainly for printing purposes. Through rendering engines that process these directives, it can publish to a variety of formats, including PDF.
  - XML Compliance Form data transfer file: This will append an embedded encrypted pdf of the compliance form.
- Next few slides are from a prototype of the Compliance Forms Engine developed by Robert Scott on the Residential ACM team

BEES Reporting Engine Process Prototype

[Log In]

Home | Notes | About

BEES Report Processing - Entry Point

The Entry Point for the BEES Reporting Engine is a successful BEES Compliance Analysis, represented by well-formed xml data that is validated against a common schema document. This demonstration uses an xml schema derived from a prototype 2008 BEES Prescriptive CF1R data model as it's source. While this application uses a web browser user interface, the implementation would be a web service or similar server process and would allow for interoperability between a variety of programming platforms for third-party software.

In this initial step, the submitted xml data is initially validated against the current official schema in a datastore. The result of the validation will return a status and if successful, allow proceeding to the next step in the workflow.

Upload a file to the service. A zip file of valid and invalid files for testing can be downloaded from here

Select

Submit

BEES Report Processing - Data transformation

The valid xml data is now transformed using a common xslt style sheet that includes html for screen display. In a real-time environment this step could be used to verify the data prior to submission to the next step of document formatting and output.

Select next when ready.

Next

An XSL-FO document similar to the XSLT Style Sheet used in the transformation step is applied to the XML Data to generate the formatted report. XSL-FO stands for Extensible Stylesheet Language Formatting Objects, and is a web standard. This requires a, XSL-FO Processor, which is the software required for formatting xsl documents for out-put. This demonstration uses a port of the Apache FOP 2.0 (Formatting Objects Processor) implemented as server-based code.

Click the button below to generate a pdf document using the xml data and xsl-fo document from the above links.

Generate PDF

BEES Report Processing - Embed Document Image into XML Data

The document created in the previous step can now be embeded into same xml data and returned as a single package to the client application. The pdf is changed into into ASCII bytes using a relatively simple binary encoding algorithm and inserted as an element in the xml. The document can then be retrieved by decoding using the same version encoder.

Insert PDF Data

BEES Report Processing - Conclusion

The following list of elements comprise the BEES Reporting Engine:

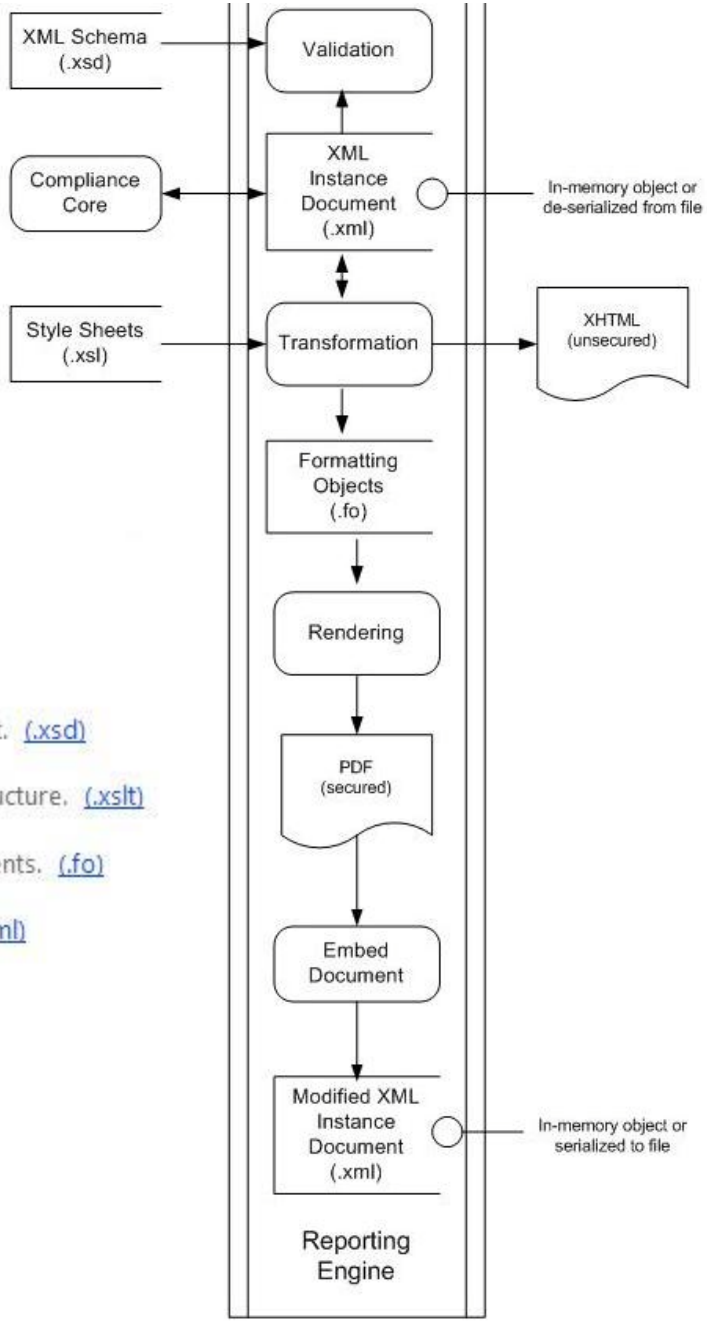
A standardized xml data document that can both be consumed by and output from compliant BEES software. (.xml)

A common schema that defines the data model represented by the xml data document. (.xsd)

A standard definition of how these data are transformed into a standard reporting structure. (.xslt)

A standard definition of how these data are formatted for the standard report documents. (.fo)

The content returned by the reporting engine is appended to the same input data. (.xml)



Compliance Forms

- Questions?

Pilot Projects

- 3rd Party Adoption of the Compliance Engine

- Alternative User Interfaces
- Alternative Software API
- Alternative Simulation Engines

- Diagram

ACM Reference Method

- **Software Testing Procedure**
- Comparative tests to determine whether applicant software programs are acceptable for use in code compliance
- Results from applicant software will be compared to “reference” results (generated using EnergyPlus)

ACM Reference Method

- **Scope of Tests:**
  - Opaque Envelope
  - Glazing
  - Overhangs
  - Occupancy
  - Lighting
  - Daylighting
  - Plug loads
  - Other Process loads
  - Schedules
  - HVAC systems
  - HVAC controls
  - Central Plant systems
  - Domestic hot water systems
  - Renewable energy systems

ACM Reference Method

- **Test Models**
  - Will be based on a subset of the DOE Prototype Building Models
    - Office, Retail, School, Warehouse, Restaurant
    - Hot, Cold, and Mild Climate Zones
  - Models will be simulated
  - One or more parameters will be modified for subsequent tests to cover the scope of tests
  - Suites of tests will be defined to cover each group of design strategies

ACM Reference Method

- **Test Models**
  - Refer to Spreadsheet

ACM Reference Method

- **Compiling the Results**
  - Annual site energy consumption
  - Annual TDV
  - End-Use breakdowns
  - Comfort metrics
  - % variance between test cases

ACM Reference Method

- **Determining Passing Criteria for Applicant Software Programs**
  - To be considered – how to judge alternate simulation engines
  - % variance metric may be more critical than absolute energy consumption metrics

ACM Reference Method

- Questions?

Wrap-Up

- Documents to be distributed as follow up include:
  - Software Functional Requirements
  - ACM Reference Method Summary
  - Summary of discussion points from today's meeting
- We will seek further input on key items
- Timeframe for next PAC meeting

Program Advisory Committee

Thanks for Your Participation!