Stage 4

REPORT DATE OF PUBLICATION

Technical Memorandum: UCPRC-TM-2018-04

Draft

eLCAP: A Web Application for Environmental Life Cycle Assessment of Pavements

Authors:
Jon Lea, John Harvey, and Arash Saboori

Funded by Partnered Pavement Research Center (PPRC) Project Numbers 4.54 (DRISI Task 2718, "Environmental Life Cycle Assessment Updates and Applications") and 4.66 (DRISI Task 3191, "Environmental Life Cycle Assessment Updates and Applications"), and with funds from the University of California Pavement Research Center

PREPARED FOR:

California Department of Transportation
Division of Research, Innovation, and System Information

PREPARED BY:

University of California
Pavement Research Center
UC Davis, UC Berkeley

TECHNICAL REPORT DOCUMENTATION PAGE

1. REPORT NUMBER UCPRC-RR-2018-04	2. GOVERNMENT ASSOCIATION NUMBER	3. RECIPIENT'S CATALOG NUMBER
4. TITLE AND SUBTITLE eLCAP: A Web Application for Environmental Life Cycle Assessment of Pavements		5. REPORT PUBLICATION DATE TBD 6. PERFORMING ORGANIZATION CODE
7. AUTHOR(S) J. Lea (ORCID 0000-0003-0999-469X), J. Harvey (ORCID 0000-0002-8924-6212), and A. Saboori (ORCID 0000-0003-0656-8396)		8. PERFORMING ORGANIZATION REPORT NO. UCPRC-RR-2018-04 [ITS-D number to come]
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of California Pavement Research Center Department of Civil and Environmental Engineering, UC Davis 1 Shields Avenue Davis, CA 95616		10. WORK UNIT NUMBER 11. CONTRACT OR GRANT NUMBER 65A0542 and 65A0628
12. SPONSORING AGENCY AND ADDRESS California Department of Transportation Division of Research, Innovation, and System Information P.O. Box 942873 Sacramento, CA 94273-0001		13. TYPE OF REPORT AND PERIOD COVERED 2014 to 2017 14. SPONSORING AGENCY CODE
15. SUPPLEMENTAL NOTES [DOI to come]		

16. ABSTRACT

The California Department of Transportation (Caltrans) has a growing need to be able to quantify its greenhouse gas (GHG) emissions and the other environmental impacts of pavement operations, and to consider GHG and those other impacts in pavement management, conceptual design, design, materials selection, and construction project delivery decisions. Caltrans also needs to be able to evaluate the life cycle environmental impacts as part of policy and standards development. All of these tasks can be performed using life cycle assessment (LCA), although there are different constraints and requirements with respect to the scope of the LCA and the data available for each of these different applications. The web-based software *environmental Life Cycle Analysis for Pavements (eLCAP)* is a project-level LCA tool that uses California- and Caltrans-specific life cycle inventories (LCI) and processes. The LCI database has been critically reviewed by outside experts following ISO standards. *eLCAP* models the life cycle history of a pavement project by allowing a user to specify any number of construction-type events, occurring at a user-specified date, followed by an automatically generated Use Stage event that begins immediately afterward and lasts until the next construction-type event or the End-of-Life (EOL) date. The Use Stage models currently consider the effects of roughness in terms of International Roughness Index (IRI) and use the same performance models that are used in the Caltrans pavement asset management system software, *PaveM. eLCAP* performs a formal mass-balancing procedure on a pavement LCA project model and then computes 18 different impact category values, including Global Warming Potential (GWP), Human Health Particulate Air, Acidification, Primary Renewable Energy, and others, and generates a detailed *Excel*™ report file to display graphs and tables of results. The results can be presented in terms of life cycle stage, material types, and other details.

17. KEY WORDS life cycle assessment, environmental life cycle assessment LCA, life cycle inventory, greenhouse gas, use stage, GHG, web application, cradle-to-gate	18. DISTRIBUTION STATEMENT No restrictions. This document is available to the public through the National Technical Information Service, Springfield, VA 22161	
19. SECURITY CLASSIFICATION (of this report) Unclassified	20. NUMBER OF PAGES 81	21. PRICE None

Reproduction of completed page authorized

UCPRC ADDITIONAL INFORMATION

1. DRAFT STAGE Stage 4	2. VERSION NUMBER 1
3. PARTNERED PAVEMENT RESEARCH CENTER STRATEGIC PLAN ELEMENT NUMBERS 4.54 and 4.66	4. DRISI TASK NUMBERS 2718 and 3191
5. CALTRANS TECHNICAL LEAD AND REVIEWER(S) Deepak Maskey	6. FHWA NUMBER TK
7. PROPOSALS FOR IMPLEMENTATION NA	
8. RELATED DOCUMENTS	
9. LABORATORY ACCREDITATION The UCPRC laboratory is accredited by AASHTO re:source for the tests listed in this report	AASHTO ACCREDITED

10. SIGNATURES

J. Lea **FIRST AUTHOR**	J.T. Harvey **TECHNICAL REVIEW**	D. Spinner **EDITOR**	J.T. Harvey **PRINCIPAL INVESTIGATOR**	D. Maskey **CALTRANS TECH. LEAD**	T.J. Holland **CALTRANS CONTRACT MANAGER**

TABLE OF CONTENTS

LIST OF FIGURES ..... **vii**
LIST OF TABLES ..... **viii**
PROJECT OBJECTIVES ..... **x**
1 OVERVIEW **1**
1.1 Need for *eLCAP* 1
1.2 Overview of *eLCAP* 1
1.3 Basic Results 3
1.4 Users 3
1.5 Future Directions 4
1.6 Software Ownership, Hosting, and Management 4
1.7 A Note on this Technical Memorandum 4
2 BALANCING, MODEL GENERATION, AND ASSESSMENT **5**
2.1 Balancing 5
2.2 Process Data 7
2.2.1 Description and Format 7
2.2.2 GaBi-Generated Processes 8
2.2.3 Manually Created Processes 8
2.3 Flow Data 9
2.3.1 Description and Format 9
2.3.2 GaBi-Generated Flows 9
2.3.3 Manually Created Flows 9
2.4 Models 9
2.4.1 Description and Format 10
2.4.2 User-Defined Models 10
2.5 Processing and Generation of the LCA Database 11
2.6 Model Generation 11
2.6.1 Construction-Type Event 11
2.6.2 Use Stage Event 14
2.7 Assessment (LCIA) 14
2.7.1 Flow-Based Impacts 14
2.7.2 Performance Model-Based Impacts 16
3 ASSOCIATION WITH *GaBi* **23**
3.1 Process Type: “agg” 23
3.2 Process Type: “u-so” 23
4 USER INTERFACE **24**
4.1 Authentication and Authorization 25
4.2 Projects and Project Trials Management 27
4.3 Project Information 28
4.4 Life Cycle Definition 30
4.5 User-Defined Processes 35
4.6 Default/Min/Max Values 37
4.7 Analyze & Results 37
4.7.1 Results Generation 40
5 ARCHITECTURE **44**
5.1 ASP.NET Concepts 48
5.1.1 HTML (aspx, server controls, CSS), C# Code behind and Event and Error Handling 48
5.1.2 Master and Content Pages 49
5.1.3 Web.config 49
5.2 Authentication, Authorization, Profiles and the Database 50
5.3 Automatic Email Generation 50
5.4 Units 50

5.4.1 Base and Client 51
5.4.2 Per Data Item 51
5.4.3 UI and C# Code 52
5.5 User Real-Time Feedback: SignalR 52
5.6 Main Classes 53
5.6.1 Base Class 53
5.6.2 Life Cycle 53
5.6.3 Important LCA Classes 55
5.7 Third-Party Libraries 58
5.8 Application/Session Startup 58
5.9 Software Development 58
5.9.1 The Environment 58
5.9.2 NuGet 59
5.9.3 Source Control 59
6 DATA ACCESS 60
6.1 Database Connections 60
6.1.1 MS SQL Server 60
6.1.2 MS Access 63
6.2 XML Database File 65
REFERENCES 66
APPENDIX A: PROCESS MODELS 67

LIST OF FIGURES

Figure 2.1: A unit process. 5
Figure 2.2: Two-unit process with scaling. 6
Figure 2.3: Example *eLCAP* model. 10
Figure 2.4: Overall *eLCAP* data processing and operation. 12
Figure 2.5: Pavement project model with other upstream models. 13
Figure 2.6: Excerpt from the table of IRI performance model parameters. 17
Figure 2.7: Example pavement project lane configuration. 19
Figure 2.8: Sample from the *CalTrucks* traffic table. 19
Figure 2.9: Flowchart for computing ESALs/yr. 20
Figure 2.10: Plot of Caltrucks Traffic Data (directional). 21
Figure 2.11: Example Use Stage results. 22
Figure 4.1: The *eLCAP* home page. 25
Figure 4.2: Accessing the registration page. 26
Figure 4.3: Registration page. 26
Figure 4.4: Password recovery page. 26
Figure 4.5: Accessing the password change page. 27
Figure 4.6: Password change page. 27
Figure 4.7: Projects and Trials Management page. 28
Figure 4.8: The menu item that opens the Project Information page. 28
Figure 4.9: Project Information page. 29
Figure 4.10: The menu item that opens the Life Cycle page. 30
Figure 4.11: Selecting a treatment for the use stage roughness equation. 30
Figure 4.12: Life cycle definition page. 33
Figure 4.13: The page to manage user-defined processes. 36
Figure 4.14: Adding User-Defined HMA Process page. 37
Figure 4.15: Analyze & Results page. 39
Figure 4.16: Analyze & Results page showing Use Stage event summary results. 39
Figure 4.17: Generated reports ready for download. 40
Figure 4.18: Debug Level Report results for the Pavement Project process. 41
Figure 4.19: Debug Level Report results for the Use Stage. 41
Figure 4.20: *Excel* report showing bar charts. 42
Figure 4.21: *Excel* report showing 3D bar charts. 42
Figure 4.22: *Excel* report showing data table. 43
Figure 5.1: Groups of classes in the Utilities DLL. 45
Figure 5.2: Factory Pattern implemented in DAL. 46
Figure 5.3: Provider Pattern implemented in DAL. 47
Figure 5.4: The Base Class, *UCPRCBase*. 53
Figure 5.5: Life cycle classes. 54
Figure 5.6: *LcaBase*, *PavementLca*, and *PavementUseLca* classes. 55
Figure 5.7: Model and process-related classes. 56
Figure 5.8: LCIA assessment-related classes. 57
Figure 6.1: *eLCAP* database schema. 60
Figure 6.2: SQL Server project data connection string. 61
Figure 6.3: ASP.NET user authentication, authorization, and profile database schema. 62
Figure 6.4: *SQL Server* log-in connection string. 62
Figure 6.5: MS *Access* traffic database schema. 63
Figure 6.6: MS *Access* WIM database schema. 63
Figure 6.7: MS *Access* axle load spectra (24 hours) database schema. 64
Figure 6.8: MS *Access* climate database schema. 64
Figure 6.9: MS *Access* connection string. 64
Figure 6.10: XML file schema. 65

Figure A.1: Pavement Project Process Model. 67
Figure A.2: HMA Process Model. 68
Figure A.3: Bitumen (Crude Oil) Process Model. 69
Figure A.4: Portland Cement Concrete (PCC) Process Model. 70
Figure A.5: Hydraulic Cement Process Model. 71
Figure A.6: Aggregate Base (AB) Process Model. 72
Figure A.7: Crushed Stone Process Model. 73
Figure A.8: Sand & Gravel Process Model. 74

LIST OF TABLES

Table 2.1: GHG Equation Variable Definitions 16
Table 2.2: Roughness Factor and "Const" for GHG Equation 17
Table 2.3: ESAL Categories 18
Table 2.4: Climate Categories 18

DISCLAIMER

This document is disseminated in the interest of information exchange. The contents of this report reflect the views of the authors who are responsible for the facts and accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the State of California or the Federal Highway Administration. This publication does not constitute a standard, specification or regulation. This report does not constitute an endorsement by the Department of any product described herein.

For individuals with sensory disabilities, this document is available in alternate formats. For information, call (916) 654-8899, TTY 711, or write to California Department of Transportation, Division of Research, Innovation and System Information, MS-83, P.O. Box 942873, Sacramento, CA 94273-0001.

A Note on Funding

This project was funded by Caltrans DRISI through the Partnered Pavement Research Contract Strategic Plan Elements (PPRC SPEs) 4.54 and 4.66, and with internal funding provided by the University of California Pavement Research Center (UCPRC).

PROJECT OBJECTIVES

This project is part of Partnered Pavement Research Center Strategic Plan Elements (PPRC SPEs) 3.46, "Environmental Life Cycle Assessment Tool for Project-Level Use," and 3.55, "Implementation of Environmental Life Cycle Assessment (LCA) Data and Models for Project-Level Use in the eLCAP Software." The coding and software development were partially funded using internal University of California Pavement Research Center funds.

The objective of Project 3.55 is to continue the development of a web-based online pavement LCA tool that uses California-specific datasets for energy and material and that follows Caltrans construction practices. The tool will be updated using information developed by the UCPRC for Caltrans in previous projects (4.66, "Environmental Life Cycle Assessment Updates and Applications" and 4.73, "Fast Model Energy Consumption Structural Response") and the companion project in the current contract (4.80, "Environmental LCA Updates and Applications"). The tool will be consistent with the Federal Highway Administration (FHWA) Pavement Life Cycle Assessment Framework and the work of federal agencies (including FHWA) in the Federal Commons initiative.

The data and procedures in *eLCAP* will be updated for use at the conceptual-level design and project-level design stages. User interfaces and documentation for the tool will also be updated based on user feedback. *eLCAP* will also be further updated so it is compatible with how *PaveM* calculates roughness performance and GHG emissions and other LCA updates that may be made in *PaveM*. The work will include an outside critical review of the tool itself (the inventories and models will be subject to a formal outside critical review as part of Project 4.80).

The objective of Project 3.55 will be achieved by completing its following tasks:

- Task 1: Update *eLCAP* with improved and new models at every pavement life cycle stage
- Task 2: Implement a conceptual design-level module for roadway analysis
- Task 3: Update the user interface and system requirements
- Task 4: Implement *eLCAP* after review and testing by UCPRC and Caltrans
- Task 5: Submit the tool for outside critical review and respond to comments
- Task 6: Update the software, software documentation, and help system

This technical memorandum is one deliverable satisfying Task 6.

It should be noted that the *eLCAP* software discussed and depicted in this technical memorandum represented the most recent version of the software available at the time of the writing. However, as the development *eLCAP* is continual, the software's functions, user steps, and/or interface may be different at a later date.

1 OVERVIEW

1.1 Need for *eLCAP*

The California Department of Transportation (Caltrans) has a growing need to be able to quantify its greenhouse gas (GHG) emissions and the other environmental impacts of pavement operations, and to consider GHG and those other impacts in pavement management, conceptual design, design, materials selection, and construction project delivery decisions. Caltrans also needs to be able to evaluate the life cycle environmental impacts as part of policy and standards development. All of these tasks can be performed using life cycle assessment (LCA), although there are different constraints and requirements with respect to the scope of the LCA and the data available for each of these different applications.

Caltrans currently uses the *PaveM* asset management software for pavement management. This software includes models for roughness, in terms of International Roughness Index (IRI), that are used with previously developed life cycle inventories (LCI) to calculate GHG emissions at the network level for planned scenarios of treatments versus "do nothing."

Caltrans is currently also using a spreadsheet-based LCA tool from the FHWA called *ICE (Infrastructure Carbon Estimator)* to obtain an estimate for GHG production at a level "higher" than what *eLCAP* operates at for the purposes of conceptual project evaluation. *ICE* functions at the corridor or higher level with very little input by the user.

There is a need for an LCA tool that models the details of the construction and maintenance life cycle of a pavement project when a user needs more detailed environmental impact results and has the additional input data required for a more detailed analysis, either at the conceptual-design stage or later in the project-design process. In addition, there is a need for a project-level LCA tool that uses LCIs specific to the materials and equipment typically used in California and by Caltrans. *eLCAP* currently fills these needs at the project-design stage. *eLCAP* is being designed so that it can also produce concept-level evaluations with California-specific data in the future.

1.2 Overview of *eLCAP*

The web-based software *environmental Life Cycle Analysis for Pavements*, also known as *eLCAP*, is a project-level life cycle assessment tool that uses California- and Caltrans-specific life cycle inventories and processes. *eLCAP* performs a formal *mass-balancing procedure* (discussed in Section 2.1) on a pavement LCA project model, and then computes 18 different *impact category* values, among which are Global Warming Potential (GWP), Human Health Particulate Air, Acidification, Primary Renewable Energy, etc. It also generates a detailed *Excel*™ report file to display graphs and tables of results.

eLCAP models the life cycle history of a *pavement project* by allowing a user to specify any number of construction-type events, occurring at a user-specified date, followed by an automatically generated Use Stage event that begins immediately afterward and lasts until the next construction-type event or the End-of-Life (EOL) date.

Construction-type events require user input specifications for materials (e.g., hot mix asphalt [HMA], portland cement concrete [PCC], aggregate base [AB], and in-place recycled [IPR] materials) and their associated quantities, transports and their associated distances, and construction equipment (e.g., pavers, rollers, lighting) and their associated times of operation. *eLCAP* has built-in library versions for these processes based on California and Caltrans practices. These library-based processes allow a user to analyze a specific pavement project or create user-defined processes based on library versions, and then customize the amounts and sources of inputs that go into that user-defined process. For example, the library process for Electricity Grid Mix uses 43.4 percent from Natural Gas, but a user can create a user-defined Electricity Process, based on the Electricity Grid Mix library process, which instead uses 20 percent from Natural Gas. Further, any custom, user-defined process set up—either by using the “Manage User Processes” page or within a project—becomes available globally to that user for any project.

Use Stage-type events, which are automatically generated for each user-defined construction-type event, have a start date immediately after the end of the construction event and an end date specified by the user. Currently eLCAP is limited in that it only computes GHG for the Use Stage, using baseline fuel consumption for a very smooth pavement and excess fuel consumption from pavement roughness (in terms of International Roughness Index [IRI]). The tool models the environmental effects of “using” the pavement project by computing the greenhouse gases (GHG) from traffic (cars and trucks) driving over it during the time span of the Use Stage, and including the effects of increasing IRI and traffic with time.

Users interact with *eLCAP* via a web browser that accesses its user interface (UI). The main UI web page contains the controls necessary to define the life cycle of a pavement project: Construction, Maintenance/Rehab, Materials, Transport and Equipment. Data for a pavement project are grouped into a *project trial;* there can be an unlimited number of project trials for a project, and a user can have an unlimited number of projects. All user data are stored in a database, currently *SQL Server*.

In addition, a user can save the data for a project trial to a local hard disk in a “json”-formatted file. These downloaded files can act as a backup to the user database or as project documentation; they can also be uploaded to *eLCAP* for processing.

1.3 Basic Results

One of the many objectives of *eLCAP* is to make the complicated process of LCA modeling and analysis as simple as possible. Another objective is to provide specific and easy-to-understand results. To that end, *eLCAP* generates an *Excel* spreadsheet that contains bar and pie charts for 18 impact categories, broken into the following categories: Material Production, Transport, Construction Equipment, and Construction. This report is generated for each construction-type event defined in the life cycle. The *Excel* spreadsheet also contains data tables for these categories' impacts.

eLCAP generates several other, lower-level reports for each construction-type event:

- Detailed process-level results from the *balancing* operation (see Section 2.1), showing scaled input and output flows for every process in the pavement project model
- The input and output flows for the LCI for the pavement project
- The flows, the characterization factor, the LCI amount, and the resulting flow potential amounts for each impact category for each stage, e.g., Material Production, and for each impact method, e.g., *TRACI 2.1*, Primary Energy

For Use Stage events, *eLCAP* generates a detailed report containing the following information for each lane in a route segment for each year of analysis in the Use Stage duration:

- Truck lane distribution factor
- Traffic volumes (cars and trucks)
- ESALs/year (for use in selecting IRI performance model parameters)
- ESAL category (for use in selecting IRI performance model parameters)
- IRI performance model parameters
- IRI
- GHG

1.4 Users

eLCAP was designed for several classes of users:

- Caltrans pavement engineers, managers and policy personnel; with future versions intended to also be used by planners working in the conceptual-design stage
- Researchers
- Local agency personnel
- Students

1.5 Future Directions

Currently, *eLCAP* has 58 specific LCIs (exported from *GaBi)* and 43 user-addressable processes (grouped into 21 types of models, such as HMA, PCC, Electricity, Paver, Grinder, etc. See Appendix A) for construction-type events (Materials and Equipment) and a Use Stage that computes GHG as a function of IRI and traffic. The following are potential enhancements being considered *eLCAP* in the future:

- Additional materials
- Additional transports
- Additional pieces of equipment
- Use Stage to include MPD, pavement deflection, etc.
- Additional impact categories for the Use Stage
- Allow users to compare one Project Trial to another Project Trial
- Allow *eLCAP* to function at a conceptual project evaluation level similar to *ICE*

1.6 Software Ownership, Hosting, and Management

eLCAP is a MicrosoftASP.NET/C# web application owned by the Regents of the University of California. The HTML (ASPX pages) and C# source files (and other support files, such as the Highway Log) are currently hosted on the servers of the UCPRC. The contractual agreement between Caltrans and the University of California allows Caltrans to move the hosting to a Caltrans web server at any time, gives Caltrans unlimited California State Government use; and gives Caltrans the ability to modify the source code to create new state-owned software.

1.7 A Note on this Technical Memorandum

It should be noted that the *eLCAP* software discussed and depicted in this technical memorandum represented the most recent version available at the time of the writing. However, as the development *eLCAP* is continual, the software's functions, user steps, and/or interface may be different at a later date.

2 BALANCING, MODEL GENERATION, AND ASSESSMENT

eLCAP's main function is to simulate (i.e., to model) the life span of a pavement section (i.e., a project) so it can to compute the environmental effects of traffic and of construction and maintenance (Use Stage). The software does this so users can make informed decisions on the best course of action to pursue to minimize harmful environment effects and maximize pavement performance over the long term. This is important because sometimes what initially sounds like a good idea may turn out not to be when all the "upstream" activities (processes) that include the extraction and production of materials, construction equipment, and traffic over a pavement project's lifetime are taken into consideration.

2.1 Balancing

eLCAP models a real-world pavement project as a series of *unit processes* or LCIs (Life Cycle Inventories) with the output "flow" of one or more unit processes going into another unit process as an "input" flow or flows, as Figure 2.1 illustrates. A unit process generates a *unit* amount of product flow (e.g., 1 kg of HMA). Each unit process has many inputs and outputs, and the outputs can be categorized as either the main *product flow* and/or one or more *emission flows.*

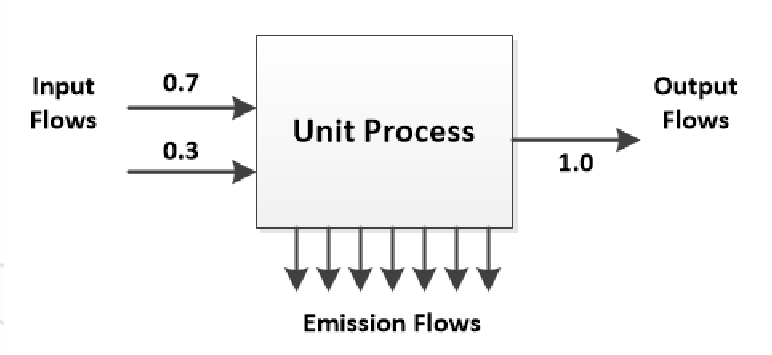


Figure 2.1: A unit process.

The *eLCAP* database currently contains over 50 different unit processes with over 1,600 different flows. A typical *eLCAP*-modeled pavement project may include several hundred unit processes for one construction-type event, and there may be many construction-type events in the overall life cycle. These collected unit processes form the *LCA balance model* for the construction-type event.

Each unit process has *input quantity* requirements. For example, an *HMA unit process* needs to have 0.06 kg of bitumen and 7.63e-3 MJ, etc., to produce 1.0 kg of HMA. Similarly, a *pavement project unit process* may need to have 100,000 kg of HMA (that is, it has an input quantity requirement of 100,000 kg of HMA).

Stage 4

Since each unit process generates a unit of product, a "scaling" or "balancing" procedure needs to be performed to get the final *balance model* in balance; this process starts at the pavement project unit process and traverses/climbs upstream for each input flow for each unit process in the model. All flows for a particular unit process are scaled by the quantity requirement of the unit process downstream. *eLCAP* accomplishes this balancing procedure by using a programming technique called *recursion.*

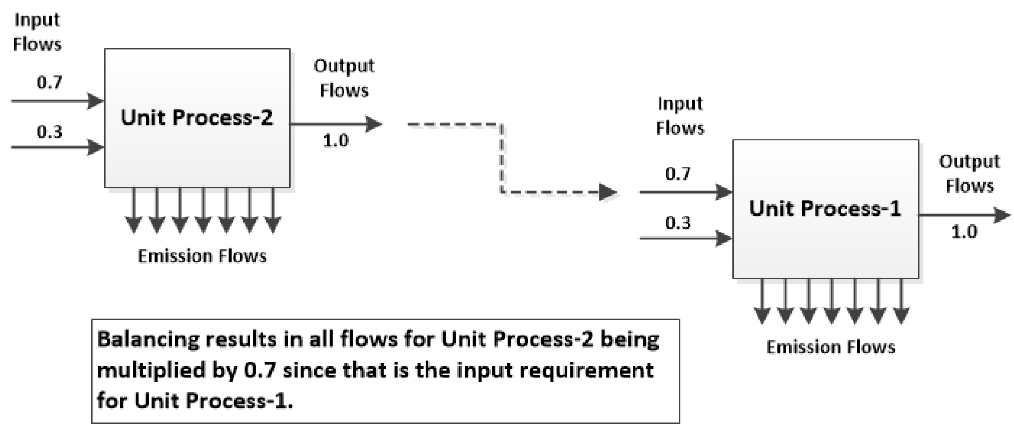


Figure 2.2: Two-unit process with scaling.

Once this balancing process is complete, each *input flow type* (e.g., CO_2) for each process is summed, and the same is done for the output flows, starting at the pavement project and traversing/climbing upstream for each input flow. The result of this summation process is an *LCI* for the pavement project that reflects all upstream effects.

The final step is to compute the results for the various impact categories. *eLCAP* computes 18 different impacts, most of which are based on *TRACI 2.1.* Each impact category consists of a list of relevant flows, each with a specific "characterization" factor (e.g., for GWP, CO_2 has a factor of 1.0 and CH_4 has a factor of 25.0). The flows for the final LCI for the pavement project are used to compute the impact results for the Construction Stage.

From the above discussion, it is evident that the building blocks of any LCA are *processes* and the *flows* into and out of those processes. Another way to state this is that the building block of any LCA is *data.* Specifically, if process-level data are representative of what is being modeled (e.g., a pavement project), then the LCA should result in a good estimate of the environmental impact of what was included in the LCA. But if the process-level data are not representative, if perhaps some of the data are for industries from a different county because "local" data are unavailable, then the LCA will not result in realistic assessments of what is being modeled.

The next sections discuss the various data items in an *eLCAP* LCA: processes, flows, and models.

2.2 Process Data

As mentioned above, the basic building block of any LCA is the *unit process*, in which there is an object that has material input needs and produces a product and emissions (i.e., output items). High-quality representative process-level data are key to obtaining representative results since LCA is basically an accounting activity, that is, data items (flows) are simply multiplied by factors and then added up.

eLCAP has over 50 unit processes in its database. This data set was created using basic LCI data from the LCA application from thinkstep AG called *GaBi (1)* and tailoring it to the California environment. The end result is that the unit processes in *eLCAP* have been designed for LCA users in California.

2.2.1 Description and Format

A file format was developed for *eLCAP* to capture the data associated with a unit process so its database can be populated. The following data items are contained in a *process definition* file*:*

- Name of the process
- Description of the process
- The kind of process
    - "u-so": a single operation process that does not reflect any upstream effects
    - "agg": a process that reflects all upstream effects
- Flows (see Section 2.3)
    - Name of the flow
    - IO type: input or output
    - Amount
    - Unit
    - If the flow is the product flow
    - Name of a parameter if the amount is arrived at via a calculation
- Parameters
    - Name of the parameter
    - Comments
    - For "Free" or constant parameters
        - Value
    - For "Fixed" or formula-based parameters
        - A formula/equation
    - Units

A separate application was created to process these *process definition* files and to insert the unit process into the *eLCAP* database. Sample lines from a process definition text file are shown below:

```
//Free parameters (constants)
Parameter, Name=Agg_Crushed, Comments="percentage of crushed aggregate", Value=47.0, Unit=%

//fixed parameters (formula based)
Parameter, Name=Agg_Crushed_norm, Formula="(Agg_Crushed/100.0) * Output_total", Comments=""

FlowItem, FlowName=Crushed stone [UCPRC Flows], IO=Input, Amount=0.0, Unit=kg,
IsReferenceFlow=false, ParameterName=Agg_Crushed_norm

FlowItem, FlowName="Hot Mix Asphalt (HMA) [UCPRC Flows]", IO=Output, Amount=1.0, Unit=kg,
IsReferenceFlow=true, ParameterName=Output_total
```

2.2.1.1 Parameters

The parameters *Free* and *Fixed* were added to the process definition file to add flexibility to them since sometimes flow amounts for a unit process are not constant but are a function of several parameters. For example, diesel consumption for a Transport is a function of mpg, max_payload, utilization_ratio, etc. A user could compute an amount to use for the flow, but it would then be impossible to change some of these parameters, which is often desirable.

Parameters are either *Free* or *Fixed*. Free-types are simply named constants to make the Process Definition file self-documenting. Fixed-types are based on a formula.

2.2.2 GaBi-Generated Processes

The majority of the Unit Processes in the *eLCAP* database are the result of their first being modeled in *GaBi*, tailored to California's needs, and then taken through an export process that generates an *Excel* file. In arriving at a unit process in this way, minor manual modifications are made to the *Excel* file which is then saved as a CSV file. As noted above, a separate application processes these *GaBi* exported files and generates process definition files. *Flow definition* files (see Section 2.3) are also generated during this processing.

2.2.3 Manually Created Processes

eLCAP also has 15 or so process definition files that are manually generated. These are necessary for unit processes that consist entirely of inputs and a single product output, such as AB (aggregate base). In this way, the manually generated unit process acts as a basic aggregator of input flows and does not generate any emissions itself.

2.3 Flow Data

Intimately associated with the unit process discussed above are *flows. Flow objects* are used to model the flows of materials and emissions. Flows connect the input items of one unit process to the outputs of another unit process.

2.3.1 Description and Format

A file format was developed for *eLCAP* to capture the data associated with a flow so the application's database can be populated. The following data items are contained in a *flow definition file:*

- Name of the flow
- Description of the flow
- Reference Quantity, e.g., mass
- Reference Unit, e.g., kg
- Flow Property (used to convert a referenced flow in a unit process to units different from those used to define the flow)

2.3.2 GaBi-Generated Flows

Flow definition files are generated from the processing of *GaBi*-exported unit-process CSV files. Currently, over 1,600 flows have been generated from the *GaBi*-export process.

2.3.3 Manually Created Flows

eLCAP also has 25 or so flow definition files that are manually generated. This is necessary for unit processes that have a manually generated process definition file since the output flow for the process, the product flow, needs to be created for the aggregator-type process.

2.4 Models

The third concept used by *eLCAP* to build its LCA database is the *model.* An example model is shown in Figure 2.3 for "Crude Oil Refinery." The figure shows that a model consists of a main unit process that produces the product ("Crude Oil Refinery (u-so)") with a series of input flows.

The input flows can be satisfied by either "agg"-type unit processes or by another model. In the figure, the "agg"-type unit processes are represented by "Crude Oil (agg)," "Transport Ocean (agg)," and "Transport Barge (agg)," which all have upstream effects in them. The other input flows in the figure are connected to the outputs of other models (i.e., "Natural Gas (Boiler), Electricity, Residual Fuel Oil, and Liquefied

Petroleum Gas).” Connecting an input flow to another model allows users to customize the upstream model in UI. Users cannot customize an “agg”-type unit process.

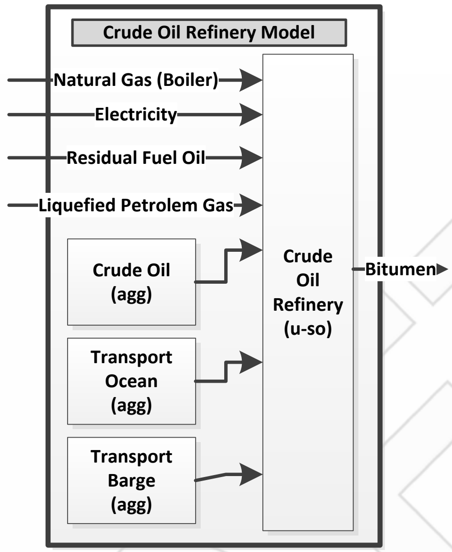


Figure 2.3: Example *eLCAP* model.

2.4.1 Description and Format

Model definition files contain *named references* to unit processes, flows, and other models. Model definition files are processed into *eLCAP* memory at program start up. The following data items are contained in a model definition file:

- Name of the model
- Description of the model
- Named references to unit processes
- For a named reference that has input items, e.g., Crude Oil Refinery (u-so) in Figure 2.3:
    - A named reference to a model
    - A named reference to a process in that model
    - A named reference to a flow in that process; this is usually the output product flow

2.4.2 User-Defined Models

All model definition files are manually generated.

2.5 Processing and Generation of the LCA Database

The left side of Figure 2.4 shows the processing procedure carried out to generate the *eLCAP* LCA database. The steps in the procedure are:

1. An LCA expert at UCPRC builds a California-specific model in *GaBi* and exports flows for the main process in the model.
2. A separate software tool, DB Gen, is used to process the *GaBi*-exported CSV files; this software tool reads these files and generates *process* and *flow definition* files.
3. DB Gen is next used again to process the generated *process* and *flow definition* text files, and also the manually generated *definition* and *flow* files.
4. The above steps result in an XML database file that *eLCAP* loads when the application starts up. It has a structure shown at the center of Figure 2.4.
5. *Model definition* files are created. All sources of LCA data are now available for *eLCAP*.
6. A user accesses *eLCAP* via a web browser. *eLCAP* reads the LCA XML database file into memory and also reads and processes the model definition files into memory.
7. The structure for the in-memory version of the XML database file mimics the structure of the XML file.
8. When a user requests that an analysis be performed, *eLCAP* builds the *balance model* (see Section 2.1), balances it, and then computes the LCI for the construction-type event and the impact factors for it.

2.6 Model Generation

When a user requests that an LCA be performed, *eLCAP* starts a loop over all the *LcaEvents* (see Section 5.6.2) in the life cycle defined by the user. And the first step in that loop, for each *LcaEvent*, is for user data to be translated into an LCA model via a *model generation activity*. This activity is simple for a Use Stage *LcaEvent* but complex for a construction-type event. Before the model generation activity, *eLCAP* must perform some initial/preparatory work to assist the Use Stage procedure: the tool converts the pavement project location on a route to a series of lane-based segments that have traffic (cars, trucks), a WIM Group, a value for ESALs, and a selected Climate for each segment. This is necessary because the Use Stage has performance models that require this data.

2.6.1 Construction-Type Event

Figure 2.5 shows part of a construction-type model used to build the *balance model*. The process starts at the *pavement project model*, considers each named reference to a process, gets the actual process from the database, copies it, and adds it to the list of processes that will become the balance model. For each input flow for the process that produces the output product, the model that is referenced is obtained and the same process is followed. This continues until there are no more upstream models to address. A recursive implementation is used for this complex procedure. Once the balance model has been constructed it can be balanced as discussed in Section 2.1.

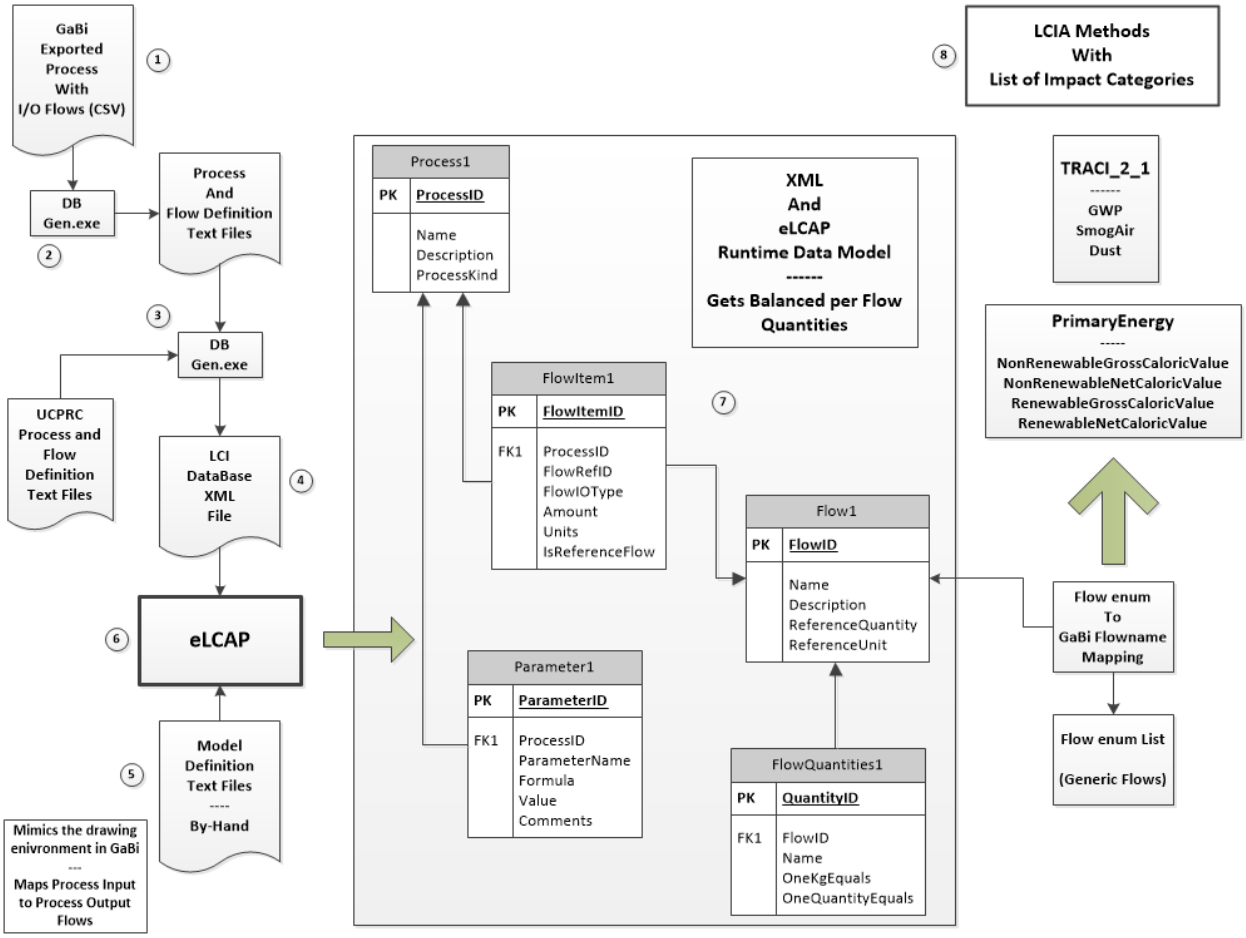


Figure 2.4: Overall *eLCAP* data processing and operation.

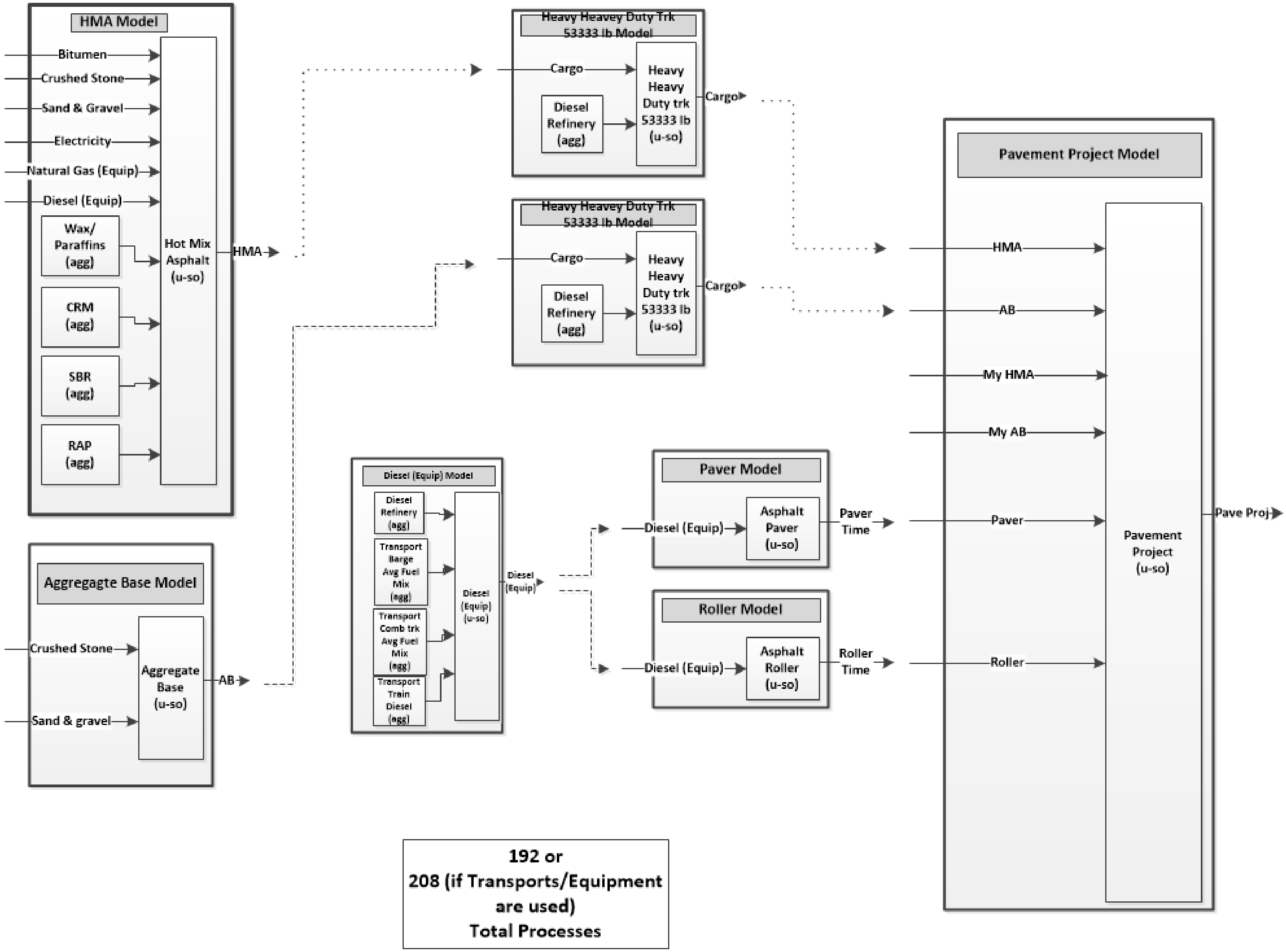


Figure 2.5: Pavement project model with other upstream models.

2.6.2 Use Stage Event

The model generation step for a Use Stage event is much simpler than for a construction-type event. The current Use Stage model computes the GHG generated from traffic driving over the pavement project for some user-defined time period. The GHG procedure used for this computation includes the effects of increasing IRI (and traffic) over time so the Use Stage model-generation step obtains appropriate IRI performance model parameters, for each lane segment, based on: Pavement Type, Treatment, Climate Zone and ESALs. See Section 2.7.2.

2.7 Assessment (LCIA)

eLCAP internally generates an appropriate set of 18 *LCA impacts* (e.g., GWP, primary energy, etc.) associated with several *LCA methods* (e.g., *TRACI 2.1)* to allow the user to make decisions on the environmental effects of a pavement project over its life cycle. No user involvement is necessary.

eLCAP includes two methods for computing impact factors:

1. A method based on a list of flows with an associated characterization factor
2. A method based on a performance model; currently, an IRI model is the only one available

eLCAP uses the first method for all impacts generated for *construction-type LcaEvents* and the second method for GHG (GWP), the only impact considered by *eLCAP* for *Use Stage-type LcaEvents.*

2.7.1 Flow-Based Impacts

As part of the *balancing* procedure in the balance model, the LCI for *appropriate processes* (see below) is generated using the methodology described in Section 2.2. The *LCI flow list* for a process and the list of *flows* (and characterization factors) for a particular impact of the process are used to compute the impact by simply matching up the flow names between the two lists, getting the balanced model LCI flow amount (which includes all upstream effects), multiplying it by a characterization factor, and summing over all flows defined for the impact.

The “appropriate processes” referenced above are those that are needed for reporting to the user, such as Construction and Material Production processes. The stages that appear in the following list (some of which are the stages are normally associated with life cycle analysis, such as Material Production, and others that are are “virtual” stages used to obtain the impacts computed for reporting purposes) are generated with the processes that have an LCI, and will be used later to compute the stages’ impacts:

1. Construction Stage
    a. The Pavement Project Process
2. Material Production Stage
    a. HMA Process
    b. PCC Process
    c. AB Process
3. HMA Production Stage
    a. HMA Process
4. HMA Aggregate Stage
    a. Coarse Aggregate Process
5. HMA Bitumen Stage
    a. Bitumen Process
6. HMA Energy Stage
    a. Electricity Process
    b. Natural Gas Equipment Process
    c. Diesel Equipment Process
7. PCC Production Stage
    a. PCC Process
8. PCC Aggregate Stage
    a. Coarse Aggregate Process
    b. Sand And Gravel Process
9. PCC Cement Stage
    a. Cement Process
10. PCC Energy Stage
    a. Electricity Process
    b. Natural Gas Equipment Process
    c. Diesel Equipment Process
11. Aggregate Base Production Stage
    a. Aggregate Base Process
12. Aggregate Base Crushed Agg Stage
    a. Coarse Aggregate Process
13. Aggregate Base Natural Agg Stage
    a. Sand and Gravel Process
14. Transport Stage
    a. Transport Process
15. Construction Equipment Stage
    a. Paver Process
    b. Roller Process

As is shown, there are 15 stages and many processes. Computing LCIs for a process is a computer intensive activity since *eLCAP* needs to recursively climb the upstream tree, starting from a particular process and considering all the input flows into that process.

2.7.2 Performance Model-Based Impacts

eLCAP computes a single impact factor for Use Stage-type *LcaEvents:* GHG. Calculating that impact factor is done using an equation that accounts for traffic loads in each *lane segment* (cars and trucks) and uses an IRI performance model that predicts the increase of IRI over time (age from the date a treatment was applied).

The equation for calculating GHG emissions from vehicles in each year is:

$$\begin{aligned} GHG_{Vehicle}(IRI) = {} & (IRI \times RoughnessFactor_{Car} + Const_{Car}) \times CarVolumeLane\, i \times LaneMile \\ & + (IRI \times RoughnessFactor_{2Axle} + Const_{2Axle}) \times 2AxleVolumeLane\, i \times LaneMile \\ & + (IRI \times RoughnessFactor_{3Axle} + Const_{3Axle}) \times 3AxleVolumeLane\, i \times LaneMile \\ & + (IRI \times RoughnessFactor_{4Axle} + Const_{4Axle}) \times 4AxleVolumeLane\, i \times LaneMile \\ & + (IRI \times RoughnessFactor_{5Axle} + Const_{5Axle}) \times 5AxleVolumeLane\, i \times LaneMile \end{aligned}$$

The emissions from this equation are summed over the analysis period (from the end of a construction-type event to the start of the next construction-type event) using the IRI predicted for the pavement at the middle of each year. The variables used in the $GHG_{vehicle}$ equation are described in the Table 2.1.

Table 2.1: GHG Equation Variable Definitions

$GHG_{Vehicle}$	GHG emissions from all vehicles in Lane *i* in a single year (metric ton)
IRI	IRI of the segment in Lane *i* (m/km), 1 m/km = 63 inches/mile
LaneMile	Total lane-miles of that lane in the segment (lane-mile)
RoughnessFactor	The averaged coefficient to convert the IRI to GHG emissions for cars and trucks due to rolling resistance. The unit is metric tonnes of CO_2-e/(IRI[m/km] x mile)
Const	Constant term in the equation, representing the GHG emissions due to other types of resistance that the energy needs to resist. The unit is metric tonnes of CO_2-e/mile.
CarVolumeLane i	Volume of cars in Lane *i*. It is readily available in the traffic table.
2/3/4/5AxleVoluemLane i	Volume of trucks in each class in Lane *i*. It is readily available in the traffic table.
$GHG_{VehicleAllYears}$	Total GHG emissions (metric tonnes) from vehicles running on Lane *i* in the segment in the whole analysis period

The values for *RoughnessFactor* and *Const* used in the $GHG_{vehicle}$ equation are shown in Table 2.2.

Table 2.2: Roughness Factor and "Const" for GHG Equation

Vehicle Classification	*RoughnessFactor*	*Const*
Car	0.00200	0.14595
2-Axle Truck	0.00063	0.24615
3-Axle Truck	0.00209	0.40898
4-Axle Truck	0.00369	0.57030
5-Axle Truck	0.00369	0.60536

The performance model equation for IRI as a function of the age of a treatment is shown below:

$$IRI(t) = a + b*age**c \qquad \textbf{[1]}$$

The parameters in Equation [1], a, b and c, are selected based on *Pavement Type* (Flexible or Rigid), *Treatment*, an *ESAL category*, and a *Climate category*. A partial table of IRI performance model parameters is shown in Figure 2.6.

				IRI (IRI_L+IRI_R)/2 = a+b*Age^c Age = ((Average_IRI-a)/b)^(1/c) (in/mi)			
Pavement Type	Treatment	ESAL/yr*	Climate**	Node ID	a	b	c
	1. Full Depth Reclamation	A	severe	1	157.3	3.4	1.0
			mild	2	88.8	2.0	1.0
		B	severe	3	143.0	5.1	1.0
			mild	4	88.3	2.2	1.0
		C	severe	5	139.6	5.4	1.0
			mild	6	87.8	2.5	1.0
	2. Thick Overlay or Reconstruct	A	severe	7	157.3	3.7	1.0
			mild	8	88.8	2.3	1.0
		B	severe	9	143.0	5.4	1.0
			mild	10	88.3	2.5	1.0
		C	severe	11	139.6	5.7	1.0
			mild	12	87.8	2.8	1.0
	3. Very Thin Overlay	A	severe	13	89.5	3.8	1.0
			mild	14	90.8	2.4	1.0
		B	severe	15	77.3	5.5	1.0
			mild	16	92.5	2.6	1.0
		C	severe	17	76.4	5.8	1.0
			mild	18	94.3	2.9	1.0

Figure 2.6: Excerpt from the table of IRI performance model parameters.

When a user defines the materials for a construction-type event, a selection is made for *Pavement Type* and *Treatment*. The other two data items needed to make the IRI model parameter selection are *ESAL Category* and *Climate Category*. Table 2.3 and Table 2.4 show how to obtain both categories from actual data.

The three ESAL categories: A, B and C, are determined by the value of ESALs/year.

Table 2.3: ESAL Categories

ESALs/yr	Category
<100,000	A
>100,000 and <500,000	B
>500,000	C

The two climate categories are determined from the *Climate Zone.*

Table 2.4: Climate Categories

Climate Zone	Category
North Coast	Severe
High Desert	Mild
Inland Valley	Mild
Central Coast	Mild
Desert	Mild
South Coast	Mild
High Mountain	Mild
South Mountain	Severe
Low Mountain	Severe

To assist in explaining the steps necessary to apply the GHG equation, the following provides an example by using a pavement section on DN-101-North, from PM R1.000 to PM R5.000. The section has the lane segment configuration shown in Figure 2.7. *eLCAP* needs to have traffic data in each *lane segment* in order to use the GHG equation.

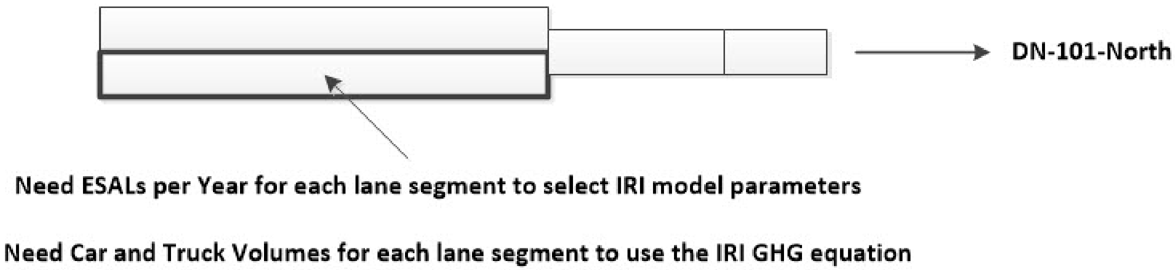


Figure 2.7: Example pavement project lane configuration.

The first step in getting the traffic data for each lane segment is to determine the *WIM Group*. This is done using the steps shown below. The WIM Group is determined for a route segment, not for an individual lane, and traffic volumes are for both directions for the route where the project is located. Traffic data comes from the *CalTrucks* database and is inflated from the date of traffic count collection to the day the *eLCAP* analysis is performed. A sample of data from the *CalTrucks* database is shown in Figure 2.8.

Determine the WIM group:

1. Get the *CalTrucks* record from *DB* (database) for *PM* (post mile) and *Leg* (e.g., in the entry "R4.569 B," B represents a Leg and means that the traffic counts are from before the PM location at the center of the segment).
2. Calculate Truck_percentage = TotalTrucks% from DB record
3. Calculate Truck_ratio = (TwoAxleVolume + ThreeAxleVolume + FourAxleVolume) / FiveAxleVolume
4. Calculate GrowthRate = TwoAxle%*7.07 + ThreeAxle%*4.8 + FourAxle%*6.27 + FiveAxle%*4.36
5. Calculate Inflated_AADT = AADT * (1.0 + GrowthRate) * (yearNow – yearCollected)
6. Use the flowchart shown in Figure 3.11 in Reference *(2)*, which uses the data items above plus some special knowledge of routes in Caltrans districts and state counties to select the WIM Group that best represents the traffic patterns for the pavement project. This means that the WIM Group selected—for example for DN-101-North—may actually be a WIM site in southern California.

Route	Route S	Distric	Cour	Pc	Postmi	Leg	AADT T	Total Tru	Total Trucks %	2 axle	2 axl	3 axle	3 axle	4 axle	4 axle	5 axle	5 axle	Description	Yea	Ve
101		1	DN	R	4.638	B	4450	507	11.4	187	36.9	93	18.4	7	1.4	220	43.3	KLAMATH, JCT.	00	V
101		1	DN	R	4.638	A	4600	474	10.3	123	25.95	63	13.29	100	21.1	188	39.66	KLAMATH, JCT.	00	V
101		1	DN		24.41	B	4600	588	12.78	168	28.57	114	19.39	79	13.44	227	38.61	SANDMINE ROA	00	V
101		1	DN		24.41	A	6300	505	8.02	152	30.1	92	18.22	61	12.08	200	39.6	SANDMINE ROA	00	V
101		1	DN		27.01	B	26500	1325	5	700	52.8	208	15.7	13	1	404	30.5	CRESCENT CITY	00	E
101		1	DN		27.01	A	11000	1056	9.6	578	54.7	175	16.6	11	1	293	27.7	CRESCENT CITY	00	E

Figure 2.8: Sample from the *CalTrucks* traffic table.

Stage 4

ESALs/year is computed for each *lane segment* using the flowchart shown in Figure 2.9 and using the axle load spectra from the Caltrans database for the selected WIM Group. The traffic volumes used to compute ESALs/year for each lane segment need to be *lane-based*, so the traffic data from the *CalTrucks* database is distributed and transformed for a specific lane. The following procedure is used:

1. *CalTrucks* traffic volumes for the segment are divided by two to get directional traffic.
2. Truck lane distribution factors are used to distribute truck traffic to each lane.
3. Car volume per lane is computed using the passenger car equivalent (PCE) approach, with a truck equal to 1.5 times a car.
4. Completing steps 1 through 3 yields final values for Lane specific AADT and Total Trucks.

Once this is complete for the lane segment, its ESALs/year can be computed using the flowchart in Figure 2.9.

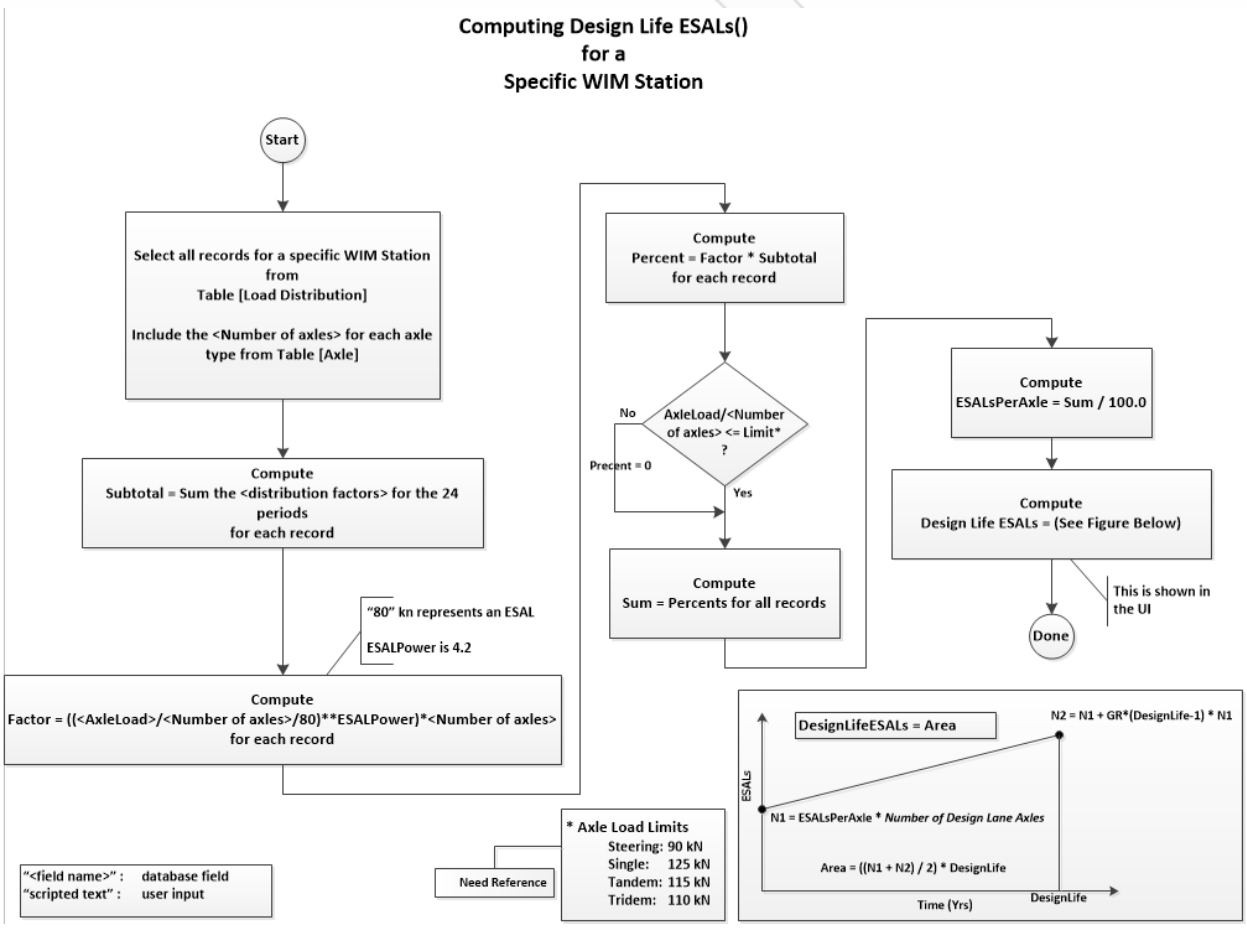


Figure 2.9: Flowchart for computing ESALs/yr.

To assist in getting traffic data into the pavement project's lane segments, *eLCAP* constructs a segment and lane configuration map (see Figure 2.7) using lane count range data from the *Caltrans Highway Log.* The traffic point list shown in Figure 2.10 (the approximate location of the example pavement project section, DN-101-North, is also shown in the figure) is constructed and then traffic data are obtained for each lane segment based on the location of the center of the segment. However, before that step is done, additional segments are added to deal with the changes in traffic data in order to avoid having a traffic value change at the middle of a segment.

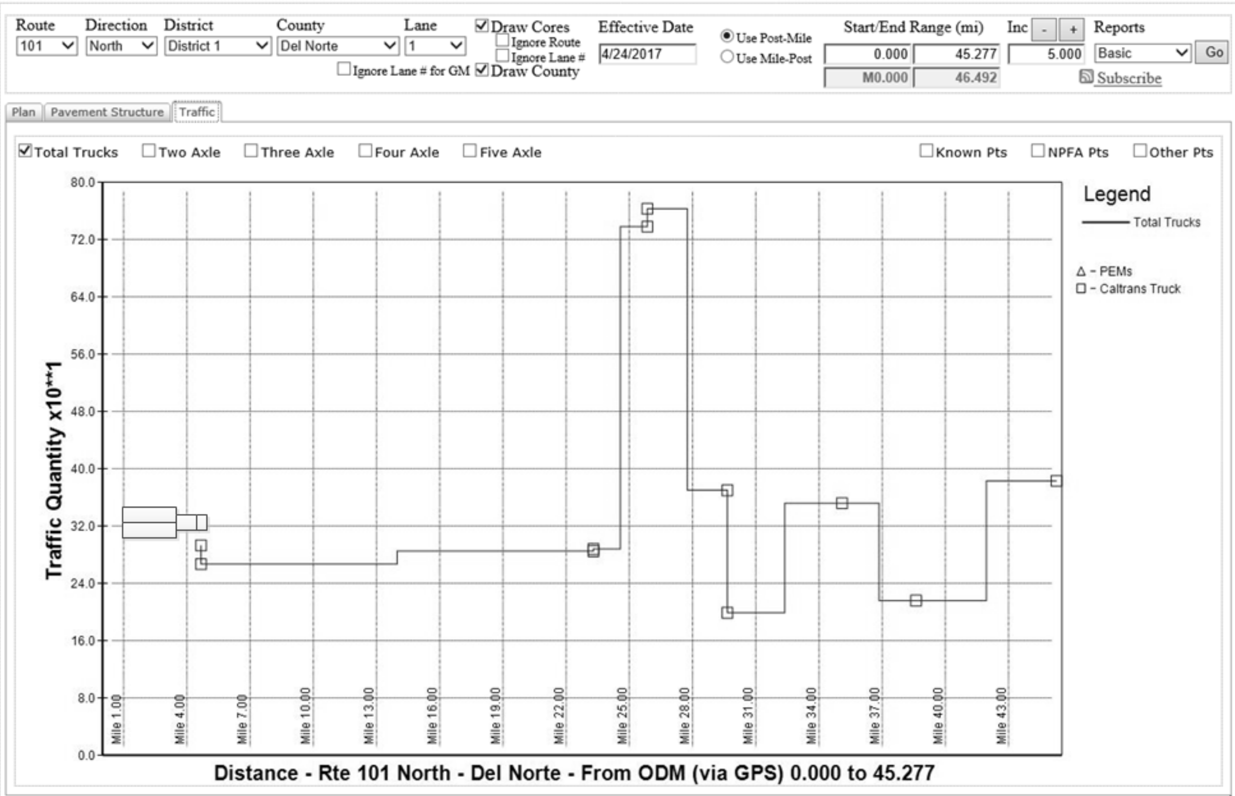


Figure 2.10: Plot of Caltrucks Traffic Data (directional).

Climate zone is obtained by using the *Caltrans Climate Zone* database, which divides the state into nine zones with Post Mile (PM) ranges for each route in each zone. The climate zone is determined for the *entire* pavement project.

Once *lane segment traffic volumes* and the climate zone have been determined, it is possible to select IRI performance model parameters using the table shown in Figure 2.6 for each lane segment.

Figure 2.11 shows some detailed *eLCAP* Use Stage results for segments and lanes in DN-101-North for the first year of a 49.54 year Use Stage.

For each route segment, the segment length, Climate category (the column labeled "C" and is "s" for "severe" for all three segments), WIM Group ("1b" for all three segments), and number of lanes per segment (two lanes in the first segment and one lane in the other two segments) are shown.

For each lane, the distribution factor, traffic volumes, ESALs/year, ESAL category (the column labeled "E"), IRI performance model parameters, IRI and, finally, the GHG are shown. Note that the ESAL Category changes between segments, "A" for the first segment and "B" for the other two; the first segment has two lanes so the traffic is spread across two lanes while the other two segments have just one lane.

```
Use Phase Data
          Starting Date: 7/1/2018    Ending Date: 1/1/2068    Analysis Period: 49.54 yrs
          Pavement Type: Flexible    Treatment: Full_Depth_Reclamation
          Project Length: 4.000 miles    Total Lane Miles: 6.647 miles    Avg # Lanes 1-Dir: 1.66

          Project Center Climate Zone: North Coast    Climate Category: severe
          Traffic Growth Rate: 0.0%


Use Phase GHG Due To IRI (Traffic Volumes Reflect Growth Rate Effects from Midpoint of Traffic Count Year to Date Shown in 'Date' Column)

Yr#    Date    Year Seg#  Seglen C WIM Ln#   DF    AADT   CarVol  TrkVol 2-Axle 3-Axle 4-Axle 5-Axle ESAL E   A     B   C   IRI       GHG
                           (mi)                                                                    (k)                  (ipm)      (kg)
--- ---------- ---- ---- ------- - --- --- ----- ------- ------- ------- ------ ------ ------ ------ ---- - ----- ---- --- ----- ------------
  1   1/1/2019  0.5    1   2.647 s  1b   1 0.190   1,152   1,104      48     17      8      0     20   23 A 157.3  3.4 1.0 159.0 4.936320E+05
                                         2 0.810   1,072     868     204     75     37      2     89   98 A 157.3  3.4 1.0 159.0 5.844774E+05
                         Segment Totals:   1.000   2,224   1,972     252     92     45      2    109                           1.078109E+06

                       2   0.991 s  1b   1 1.000   2,225   1,972     253     93     46      3    110  121 B 143.0  5.1 1.0 145.6 4.045184E+05
                         Segment Totals:   1.000   2,225   1,972     253     93     46      3    110                           4.045184E+05

                       3   0.362 s  1b   1 1.000   2,300   2,063     237     61     31     50     94  114 B 143.0  5.1 1.0 145.6 1.539020E+05
                         Segment Totals:   1.000   2,300   2,063     237     61     31     50     94                           1.539020E+05
                                                                                                             Year Total:  1.636530E+06
                                                                                              Year Total Per Lane Mile:  2.462058E+05
```

Figure 2.11: Example Use Stage results.

3 ASSOCIATION WITH *GABI*

As mentioned in Section 2.2, the majority of processes and flows in the *eLCAP* LCA database originate in *GaBi.* Models are built in the *GaBi* application using its LCIs, and then customizations are made to tailor the results for California. Last, the input and output flows for the process are exported to a file and then processed for use in *eLCAP*.

Two types of processes are exported from *GaBi:* "agg" and "u-so." These are discussed below.

3.1 Process Type: "agg"

An "agg"-type process is one that has all upstream effects included. They can be considered as "leaf" nodes in the Balance Model since nothing is upstream from them. The "Crude Oil (agg)" process in Figure 2.3 is an "agg"-type process, which has no input flows. Further, once all the input flows to an "agg"-type process have been aggregated, it is impossible to disaggregate them back into the processes used to build the "agg"-type process. An "agg"-type process, exported from *GaBi*, typically has several hundred input and several hundred output flows since all flows from upstream processes have been aggregated into it.

From a user perspective, this type of process is the least flexible since it cannot be customized. But practically speaking, there have to be "agg"-type processes at some point going upstream in the LCA model; the point where they appear depends on the sources of data and level of effort required to get the data.

3.2 Process Type: "u-so"

A "u-so"-type process is one that does not include any upstream effects; it has input flows that need to be connected to upstream models. The "Crude Oil Refinery (u-so)" process in Figure 2.3 is a "u-so"-type process. A "u-so"-type process exported from *GaBi*, typically has a handful of input flows and very few output flows: the *main product flow* and a few *emissions flows.*

A "u-so" process is more flexible than an "agg"-type since it allows flow-based connections to upstream models and thus permits an *eLCAP* user to customize the "u-so" process by changing the amounts of flows into the process and the actual source of the flows.

4 USER INTERFACE

This chapter discusses the user interface (UI) from a user's perspective; Chapter 5 discusses the UI from an architectural and software development perspective.

Figure 4.1 shows the *eLCAP* home page, which has a set of useful links and log-in controls on the left-side pane. The different pages in the UI are accessed via menu links, shown at the top in the figure. The buttons at the bottom are used to display PDF files of the various LCA models in *eLCAP*.

The following are the top menu items:

1. *Home:* this menu item is used to display the home page.
2. *Instructions:* this menu item is used to display the Instructions page, which contains information on how to use *eLCAP*.
3. *Projects:* this menu item is used to display the Projects and Project Trials Management page (see Section 4.2).
4. *Input*: this is a hierarchical menu with the following menu items:
    a. *Manage User Processes:* links to the page used to manage user-defined processes, such as a customized version of HMA (see Section 4.5).
    b. *Project Information:* links to the page used to specify the location of the pavement project on the highway system, to define the pavement cross section, and to review traffic counts (see Section 4.3).
    c. *Life Cycle:* links to the page used to define the life cycle of the pavement project, including events when construction/maintenance occurs, the list of activities, and the list of Materials and Equipment for each event (see Section 4.4).
5. *Analyze & Results:* this page is used to perform the LCA and obtain results (see Section 4.4.).
6. *Interpreting Results:* this page provides information that helps explain the results produced by *eLCAP*.
7. *About:* this menu item is used to display the About page which contains information about the application.

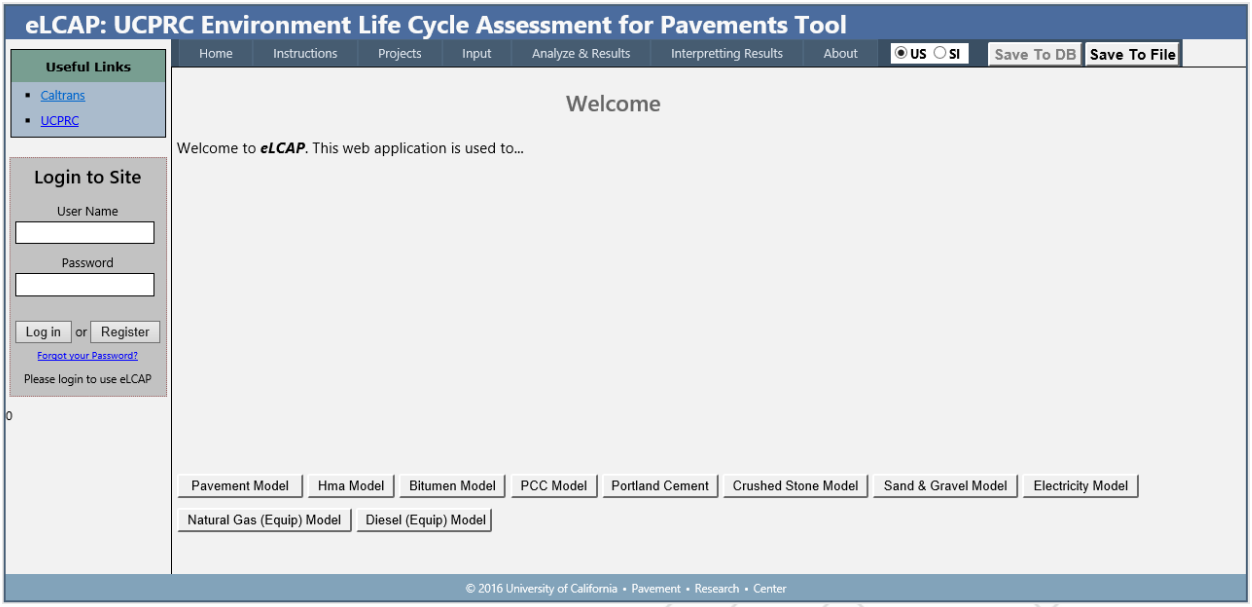


Figure 4.1: The *eLCAP* home page.

4.1 Authentication and Authorization

eLCAP requires a user to be registered and approved before they can use the program. Once approval is granted, users are associated with a particular *eLCAP* user group or groups. Specific user permissions are controlled via user groups.

The *eLCAP* registration page is accessed by clicking the “Register” button shown in Figure 4.2 (it appears on the left side of the home page). Clicking that button brings up the dialog box shown in Figure 4.3. Once a user fills in all of the empty fields and clicks “Create User,” both the user and the *eLCAP* administrator will receive an email. After receiving the email, the administrator can approve the user and assign them to the appropriate user group.

eLCAP also has a mechanism for obtaining a temporary password if a user forgets theirs. To obtain a temporary password, a registered user just has to click the “Forgot your Password?” link on the home page (as shown Figure 4.2), and this will bring up the Password Recovery page shown in Figure 4.4. After entering a user name and clicking “Submit,” the user will receive an email containing a temporary password for logging in. After doing so the user can change the password to something else, as shown in Figure 4.5. Clicking the “Change your Password” link will bring up the registration page shown in Figure 4.6.

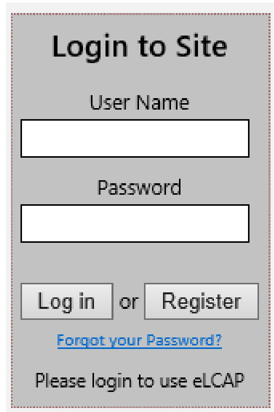


Figure 4.2: Accessing the registration page.

Figure 4.3: Registration page.

Figure 4.4: Password recovery page.

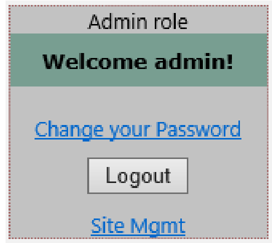


Figure 4.5: Accessing the password change page.

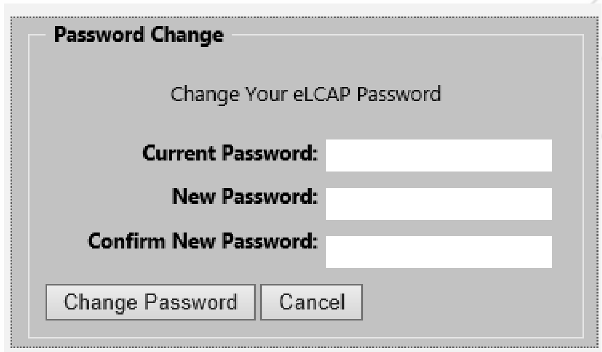


Figure 4.6: Password change page.

4.2 Projects and Project Trials Management

eLCAP uses the concepts of "projects" and "trials." A user can create any number of projects, and a project can have any number of trials. A trial contains the data for the life cycle of a pavement project. User-defined processes (see Section 4.5), such as a custom HMA or custom Transport, are common across all trials. All user data are saved in an SQL database on a server.

eLCAP allows a user to generate a text file version of Trial data to act as backup to database data or for project documentation. These files may be uploaded to *eLCAP* and used to perform an LCA as well.

The Project and Trial Management page is shown in Figure 4.7. The corresponding number-labeled areas that appear in the figure are discussed below.

1. This button allows the user to browse a local hard disk for an *eLCAP* input file, upload it, edit it, and then run it.
2. This field shows the current project.
3. This field shows the current project trial.
4. This is a link to the trial, which allows the title to be edited and a description to be entered.

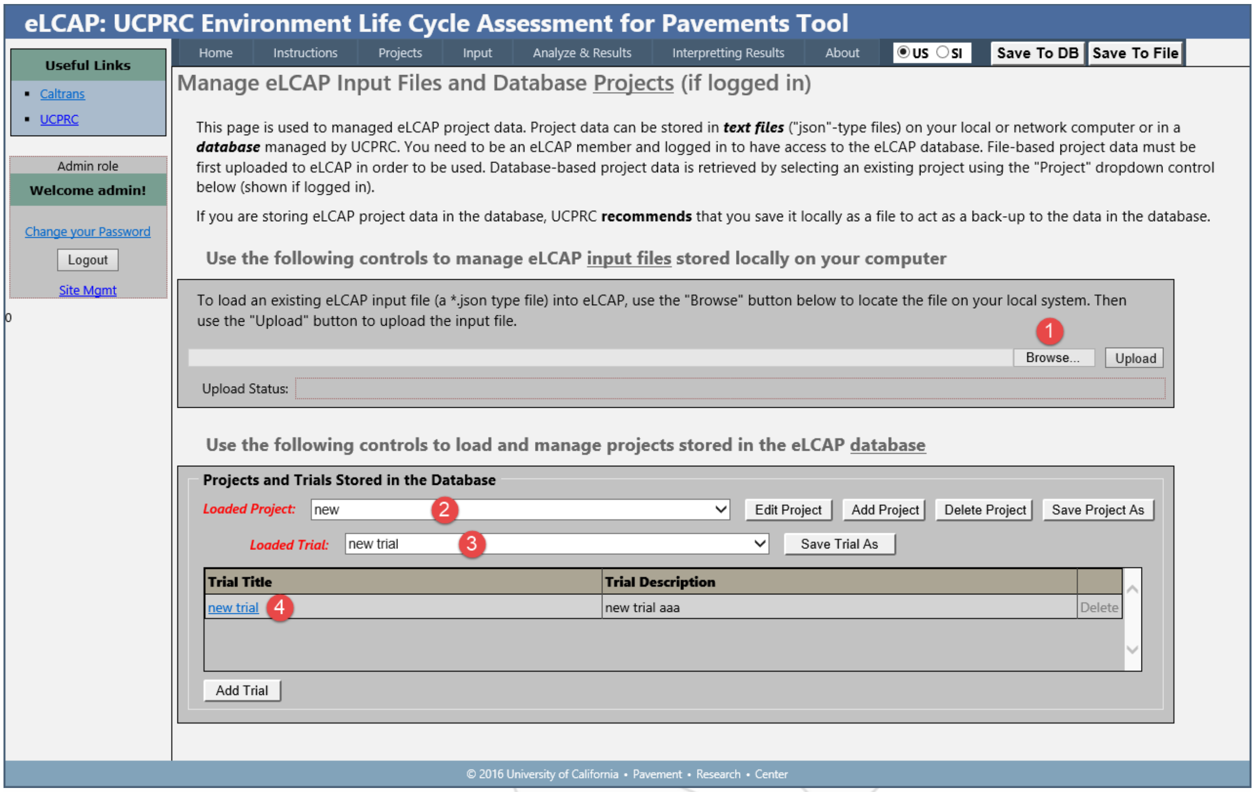


Figure 4.7: Projects and Trials Management page.

The buttons on this page allow users to perform the following activities:

1. *Edit Project:* edit the current project.
2. *Add Project:* add a new project. A new trial will automatically be added to the new project.
3. *Delete Project:* delete the current project. If there is only one project, this button is disabled.
4. *Save Project As*: creates a new project based on the current project.
5. *Save Trial As*: creates a new trial for the current project based on the current trial.

4.3 Project Information

The Project Information page is accessed via the Input menu. Hovering the mouse cursor over the Input menu will reveal three submenu items (see Figure 4.8), one of which is "Project Information."

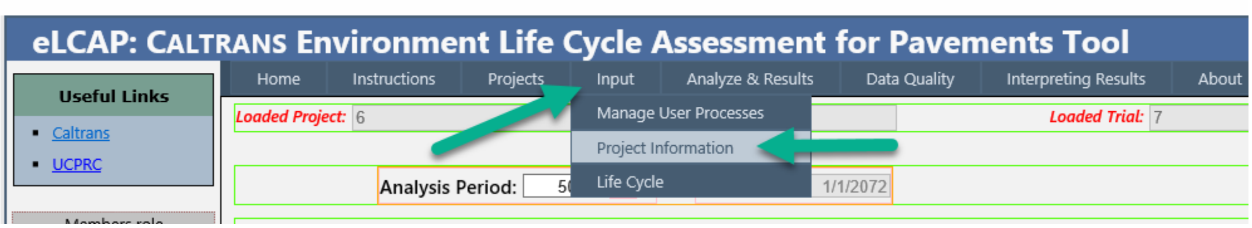


Figure 4.8: The menu item that opens the Project Information page.

Clicking Project Information will bring up the Project Information page (Figure 4.9). On this page, a user defines the location of the project and the pavement roadway cross section. The page also shows the one-way traffic counts (e.g., AADT, AADTT, etc.) for the center point of the project.

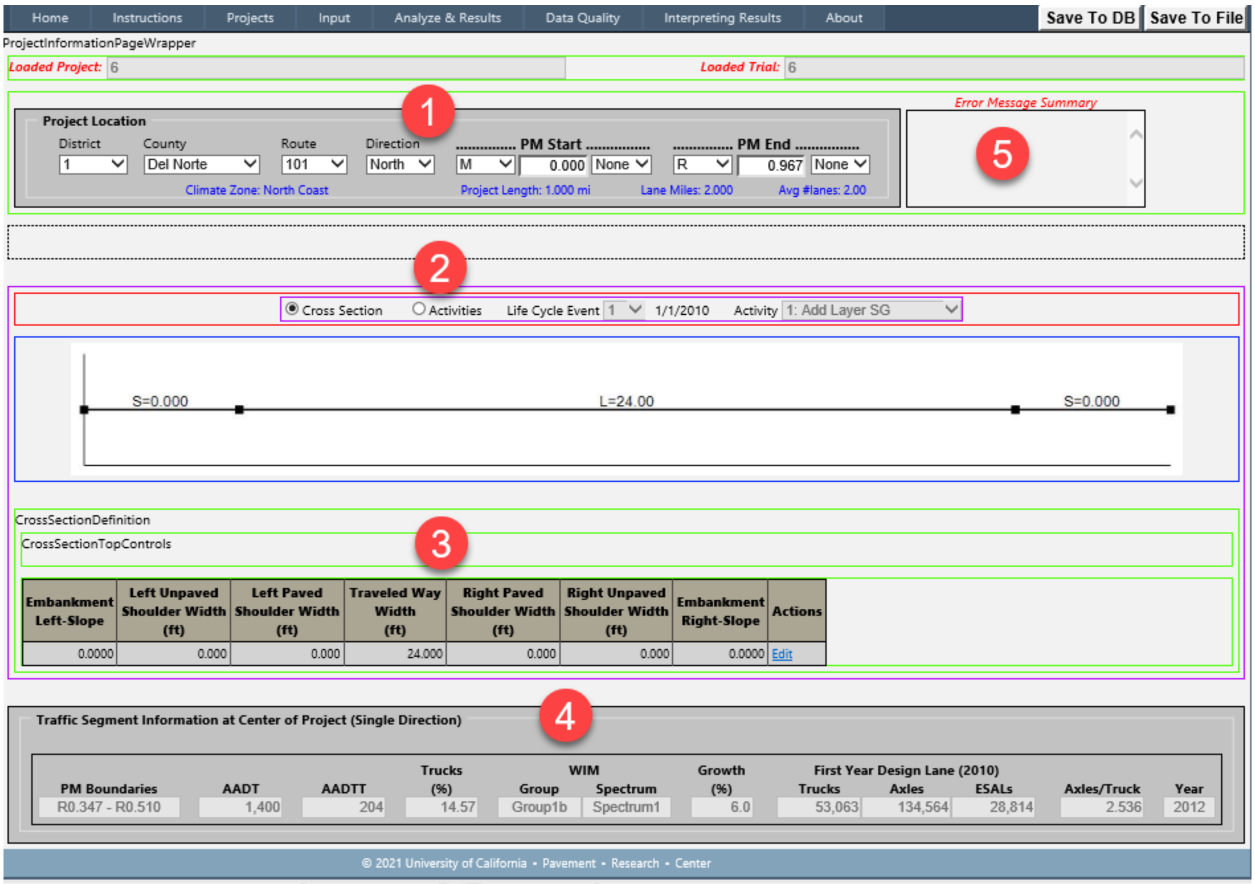


Figure 4.9: Project Information page.

The following list discusses the numbered items in Figure 4.9.

1. Controls for locating the project on the California highway system (District-County-Route-Direction and Post Mile start and end)
2. Controls for viewing the cross section or viewing activities (e.g., adding layers, removing material), per life cycle event and activity
3. A one-line table for defining the various cross-section components for the left and the right roadway side: Embankment Slope, Unpaved Shoulder Width, Paved Shoulder Width, and a single field for defining the Traveled Way Width. The traveled way is initially populated when the project location is defined; this is possible because *eLCAP* has access to the Caltrans highway log.
4. A set of read-only fields that gives the single-direction traffic counts and traffic data for the center point of the project; this is possible because *eLCAP* has access to Caltrans traffic data.
5. An error message summary area that lists all the errors found on this page.

4.4 Life Cycle Definition

The Life Cycle page is accessed via the same Input menu used to access the Project Information page (as discussed in Section 4.3). Hovering the mouse cursor over the Input menu will reveal three submenu items (see Figure 4.10), one of which is "Life Cycle."

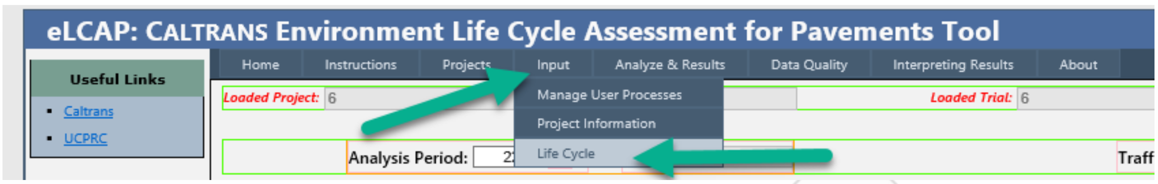


Figure 4.10: The menu item that opens the Life Cycle page.

Clicking Life Cycle will bring up the page on which a user defines the series of construction events that will model the project's life cycle. The top portion of the Life Cycle page, showing the Life Cycle grid, is shown in Figure 4.11. A Life Cycle Event is defined by the date of the event, the service life of the activities performed for the event, and the IRI roughness equation coefficients to use for the Use Stage that occurs between successive events. Each of the treatment options shown in the dropdown in Figure 4.11 has an associated set of IRI roughness equation coefficients that determines the rate of growth of IRI over time. See Section 2.7.2 for a discussion of the Use Stage. The full Life Cycle page is shown in Figure 4.12.

Selecting a treatment type for the Use Stage Roughness Equation tells *eLCAP* to do two things: to include a use stage and to select the IRI roughness equation coefficients associated with the selected treatment. By default, "None" is selected, indicating that the user does not want to include a Use Stage for the life cycle event. A user should select the treatment that best represents the end result of the set of activities defined for the life cycle event.

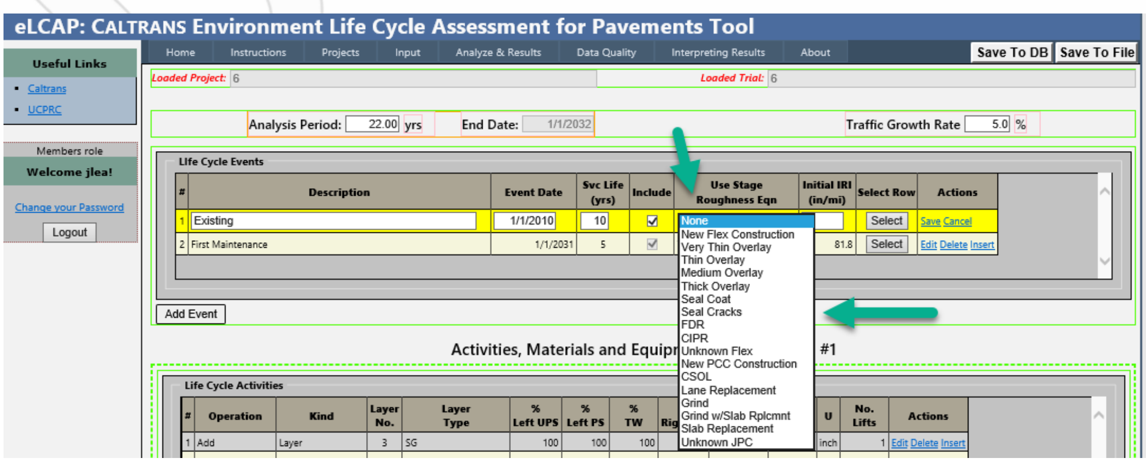


Figure 4.11: Selecting a treatment for the use stage roughness equation.

As shown in Figure 4.12, the Life Cycle page is divided into four panes:

1. *Life Cycle Events:* the grid in this pane allows a user to add any number of life cycle events. The associated activities, materials, and equipment for the selected event (highlighted in yellow in Figure 4.12) will appear in the lower three panes.
2. *Life Cycle Activities*: the grid in this pane allows a user to add any number of activities for the selected event. The following is the list of activities and the various options for each:
    a. *Add*
        i. Layer: HMA, PCC, AB, LCB, CTB-Class A, CTB-Class B, ATPB, CTPB, CCPR, FDR, PDR, AS, LTS, CSO, SG
        ii. Seal Coat: Chip Seal, Slurry Seal, Fog Seal, Cape Seal, Sand Seal, Tack Coat, Prime Coat
        iii. Reflective Coating: Bisphenol A, Polyester Styrene, Styrene Acrylate
    b. Remove
        i. Mill asphalt
        ii. Mill concrete
        iii. Grind & Groove
        iv. Cold plane
        v. Excavate
        vi. Haul Soil
3. *Materials and Transports to the Site:* the grid in this pane shows the *eLCAP*-generated materials, quantities, and means of transport for all the activities for the selected event; it also allows a user to manually add and delete materials.
4. *Equipment Used at the Site:* the grid in this pane shows the *eLCAP*-generated equipment and time of operation for each, for all activities for the selected event; it also allows a user to manually add and delete equipment.

When a life cycle event is added, the activities, materials, and equipment grids will be empty. *eLCAP* uses a default duration of 10 years between successive events to assist in constructing a life cycle. Users can edit the dates as necessary.

Each of the grids on the life cycle definition page consists of rows and columns. When the page first opens, the rows appear in *display* mode, allowing data items to be viewed but not edited. To enter *editing* mode to make changes, the user must click the "Edit" link in a particular row. After making changes, clicking "Save" will keep the changes and clicking "Cancel" will discard them.

Stage 4

When an activity is defined and saved, *eLCAP* will generate a default material to be used (e.g., a specific kind of HMA when adding an HMA layer) and a list of the construction equipment necessary to implement the activity. *eLCAP* will also compute the quantities of material using the project limits (post mile start/end and number of lanes), the cross section defined on the Project Information page, and the thickness (add layer or remove material) specified for the activity. *eLCAP* will also provide construction time estimates for each piece of equipment. The default material and the computed material quantities and equipment time estimates may be edited.

A user can delete any or all of the items generated for an activity. In addition, a user can manually add materials and equipment. In fact, the user can skip defining any activities at all and directly add materials and their associated quantities, and equipment and its associated times of operation. But in most cases, defining activities for an event is more efficient than manually building lists of materials and equipment.

Clicking the links in the Source Name column in the two lower grids—Materials and Transports to the Site and Equipment Used at the Site—will display a form page with the data for the item. For example, if a user clicks the “HMA with 15% Binder Replacement, no Rejuv” in the Source Name column of the Materials and Transports to the Site grid, the HMA form page will be displayed. Listed on that form page will be all the input flows and their associated quantities needed to produce 1 unit of HMA. If the HMA is an *eLCAP* library material, no changes can be made to it, but if the HMA is a user-defined/custom HMA mix, changes can be made.

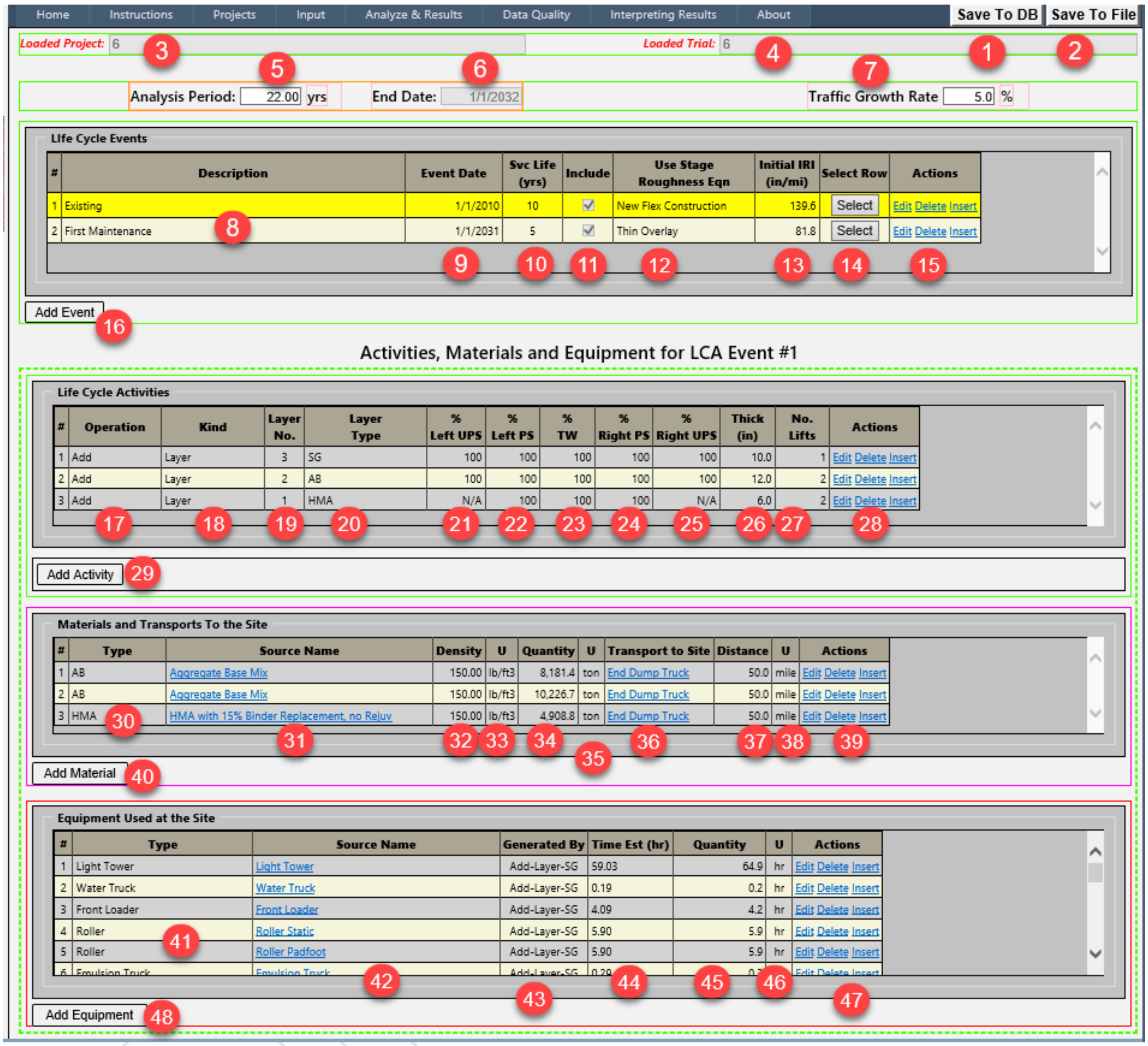


Figure 4.12: Life cycle definition page.

The following list discusses the numbered items in Figure 4.12.

1. Button to click for saving the trial data to the database. Data are saved automatically when the user goes to the "Analyze & Results" page.
2. Button to click to save the trial data to a file on the computer.
3. Field that shows the current project (i.e., the "Loaded Project").
4. Field that shows the current trial (i.e., the "Loaded Trial").
5. Field to set the analysis period.
6. End date, generated by *eLCAP*.
7. Field to set the traffic growth rate (traffic counts are used between events in the Use Stage).
8. Field to enter text that describes what the event is modeling in the life cycle.
9. Field to set the date for the event.
10. Field to set the service life of the event in years.
11. Checkbox used to tell *eLCAP* if the event should be included in the LCA.
12. Dropdown list to select the treatment type that best represents the end result of the event.
13. Field to change the default Initial IRI for the event.
14. Used to select the event that will have its activities, materials, and equipment lists displayed in the lower three grids.
15. Buttons used to edit, delete, or insert an event.
16. Button used to add an event.
17. Dropdown list to select the operation for the activity (e.g., Add/Remove).
18. Dropdown list to select the kind of operation for the activity (e.g., Layer).
19. *eLCAP*-generated layer number.
20. Material type of the layer.
21. Percent of Left Unpaved Shoulder (UPS) width to include for the activity.
22. Percent of the Left Paved Shoulder (PS) width to include in the activity.
23. Percent of the Traveled Way (TW) width to include in the activity
24. Percent of the Right Paved Shoulder (PS) width to include in the activity.
25. Percent of the Right Unpaved Shoulder (PS) width to include in the activity.
26. Thickness of the layer or the amount to remove in the activity.
27. *eLCAP* pre-populates this with the number of lifts required, per Caltrans rules.
28. Buttons used to edit, delete, or insert a life cycle activity.
29. Button to add a life cycle activity.
30. Shows/selects the type of material for the activity.
31. Shows/selects the source of the material; also used to select a custom material.
32. Field that is pre-populated with the density of the layer material.
33. Dropdown list to select the units for the density of the material.

34. Shows/selects the amount of the material.
35. Shows/selects the measurement units associated with the amount.
36. Show/selects the transport for bringing the material to the project.
37. Shows/selects the distance for the transport of the material to the project site.
38. Show/selects the measurement units for distance.
39. Buttons used to edit, delete, or insert a material.
40. Button used to add a new material to the material grid.
41. Shows/selects equipment type to be used at the site.
42. Shows/selects the piece of equipment to be used.
43. *eLCAP*-generated to indicate which activity generated the equipment.
44. *eLCAP*-generated estimate of how much equipment time will be needed.
45. Shows/selects the amount of time (U) that the equipment at the site will operate.
46. Shows/selects the measurement units for the amount of time.
47. Button used to edit, delete, or insert a piece of equipment.
48. Button used to add a piece of equipment.

4.5 User-Defined Processes

eLCAP has many built-in materials, also referred to as *library materials*, that a user can select to define a construction-type event. The library materials are organized by type, e.g., HMA, PCC, etc., and there are several individual processes per type. For example, there are three different types of HMA on which to base a user-defined, custom HMA.

Figure 4.13 shows the page for managing user-defined processes. The "Add New" button is used to add a new user-defined process for the type of material shown in the drop-down to the left of the button. The grid lists all user-defined processes.

The link in the Source Name column will display the edit form page for the process. The # Refs column indicates how many times the user-defined process is referenced by any trial. If the user-defined process is referenced, it cannot be deleted and the "Delete" button will be disabled.

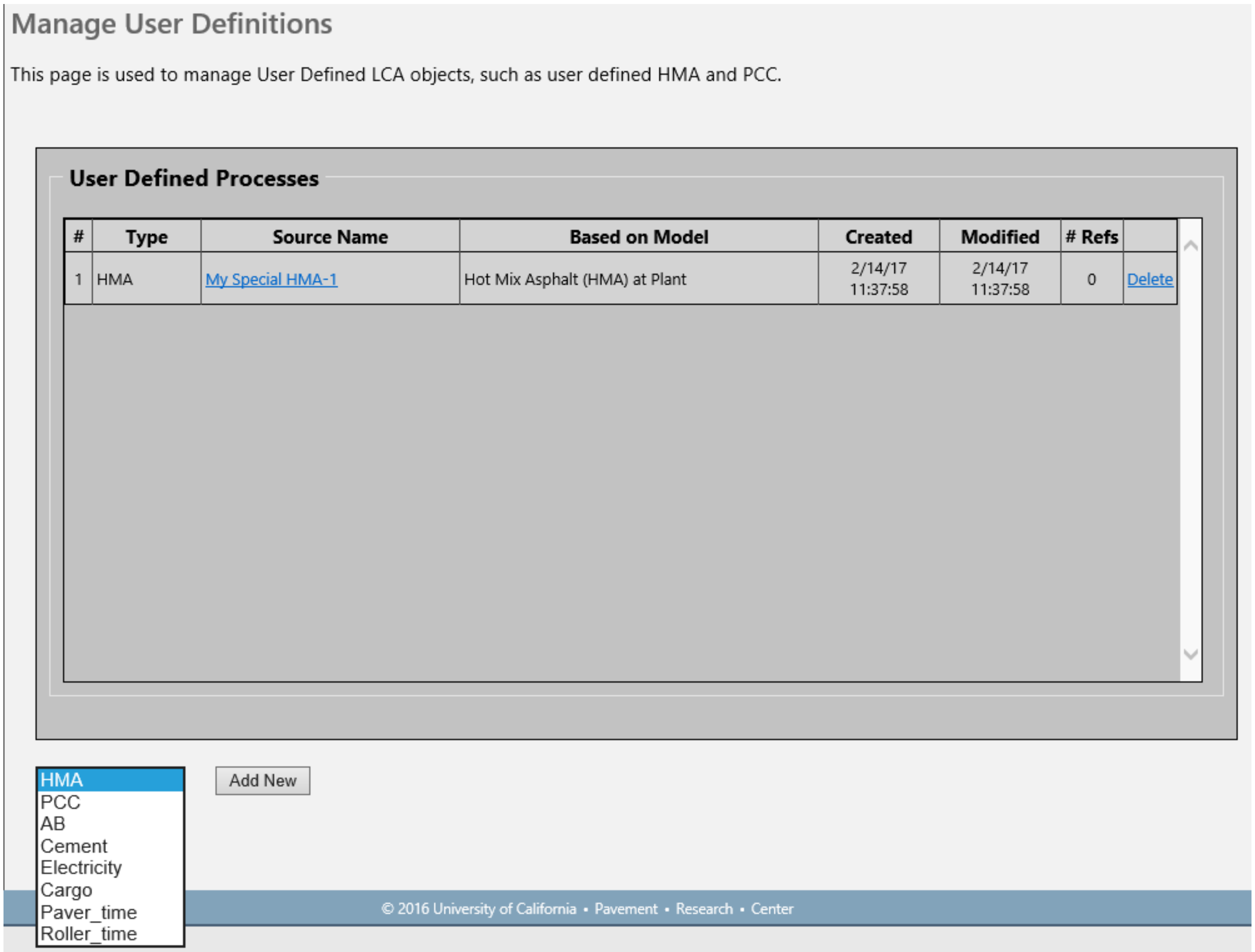
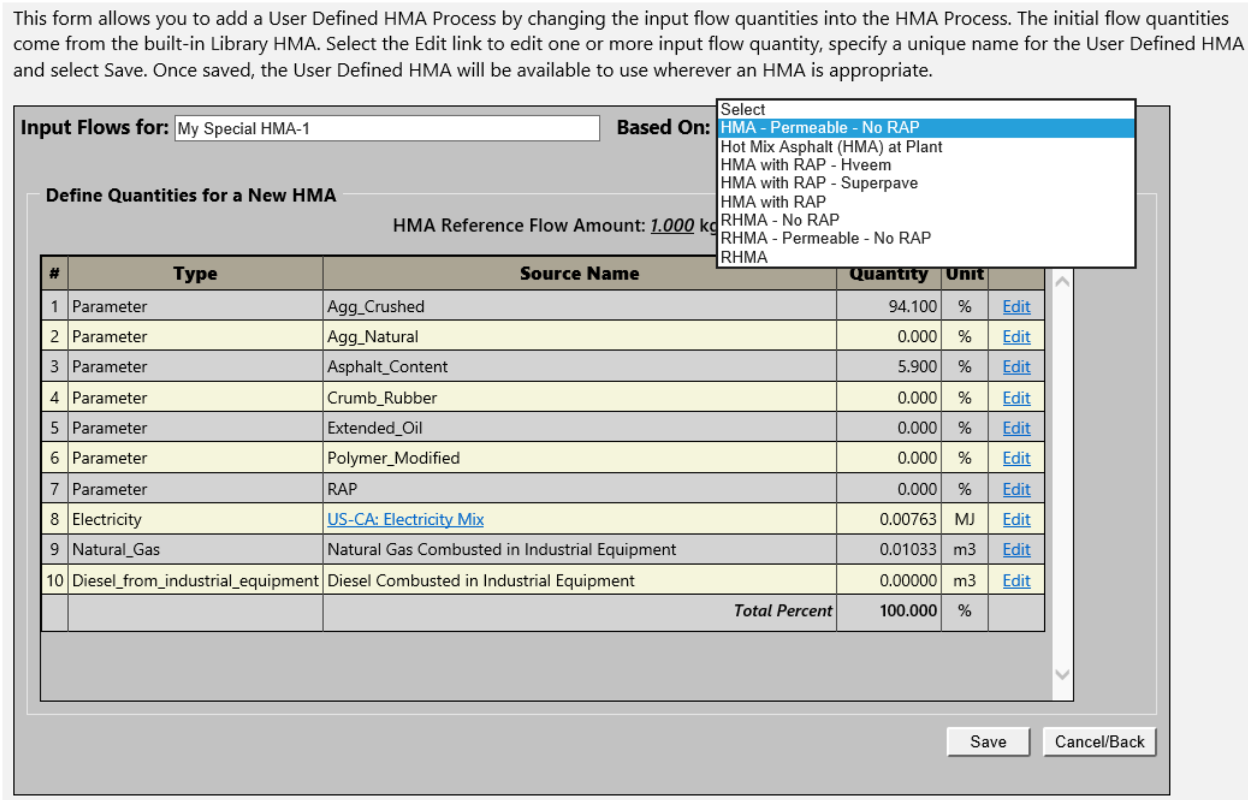


Figure 4.13: The page to manage user-defined processes.

Clicking the "Add New" button when HMA is shown in the dropdown menu brings up the HMA edit form page shown in Figure 4.14. Once this page appears, the user supplies a unique name to the user-defined HMA and then selects one of the HMA types in the *eLCAP* library by using the "Based on" drop-down menu. Next, the user edits individual rows to customize the HMA. For example, a user might want to have a special HMA that uses 4 percent "Asphalt Content" instead of the library's version of 6 percent, or maybe they want to use a user-defined process (previously defined) for electricity for this new user-defined HMA.

Once a user-defined process has been created, it may be referenced when constructing the life cycle (see Section 4.4).



This form allows you to add a User Defined HMA Process by changing the input flow quantities into the HMA Process. The initial flow quantities come from the built-in Library HMA. Select the Edit link to edit one or more input flow quantity, specify a unique name for the User Defined HMA and select Save. Once saved, the User Defined HMA will be available to use wherever an HMA is appropriate.

Input Flows for: My Special HMA-1 **Based On:**
Select
HMA - Permeable - No RAP
Hot Mix Asphalt (HMA) at Plant
HMA with RAP - Hveem
HMA with RAP - Superpave
HMA with RAP
RHMA - No RAP
RHMA - Permeable - No RAP
RHMA

Define Quantities for a New HMA

HMA Reference Flow Amount: 1.000 kg

#	Type	Source Name	Quantity	Unit	
1	Parameter	Agg_Crushed	94.100	%	Edit
2	Parameter	Agg_Natural	0.000	%	Edit
3	Parameter	Asphalt_Content	5.900	%	Edit
4	Parameter	Crumb_Rubber	0.000	%	Edit
5	Parameter	Extended_Oil	0.000	%	Edit
6	Parameter	Polymer_Modified	0.000	%	Edit
7	Parameter	RAP	0.000	%	Edit
8	Electricity	US-CA: Electricity Mix	0.00763	MJ	Edit
9	Natural_Gas	Natural Gas Combusted in Industrial Equipment	0.01033	m3	Edit
10	Diesel_from_industrial_equipment	Diesel Combusted in Industrial Equipment	0.00000	m3	Edit
		Total Percent	100.000	%	

Save Cancel/Back

Figure 4.14: Adding User-Defined HMA Process page.

4.6 Default/Min/Max Values

eLCAP has a few default values that minimize user time when creating a new project trial. The following are the default values:

- Years added to the start of first life cycle event to establish the end-of-life date: **50 years**
- Construction time duration for a life cycle event: **6 months**
- Use Stage time duration between construction events: **10 years**
- Traffic Growth Rate: **0.0%**

eLCAP also makes range checks on data items so that unreasonable numbers/results are avoided. The minimum value is usually "0.0" and the maximum value is usually a large number so as not to constrain the user too much.

4.7 Analyze & Results

The Analyze & Results page is used to perform an LCA and to obtain results. Figure 4.15 shows the Analyze & Results page. During a life cycle analysis, total GHG is displayed in the graph as it is computed for each construction-type event and each Use Stage event. Summary results appear in a scrollable window after the LCA is completed: there is a summary section for each construction-type event (Figure 4.15) and each Use Stage event in the life cycle (Figure 4.16). Report files *(Excel* and *PDF)* can be downloaded to a local computer by clicking the “Get Results” button.

The read-only fields at the top indicate the Project and Trial that will be used for the LCA.

The Balance Selection Controls are for expert users and do not appear for most users. The Model drop-down menu is used to select the LCA model that will be used for the balancing, and the Process drop-down menu is used to select the process in the model that will be the starting point for balancing. The default selection for the model is “Pavement Project” and the default selection for the process is “Pavement Project (u-so).” See Section 2.1 for a discussion of balancing.

Balancing for a process (i.e., starting at and traversing/climbing upstream from the process) results in the generation of the LCI for the process. Expert LCA users may want to generate the LCI for a different process in the Pavement Project LCA model other than the Pavement Project process and these controls allow that to be done.

To initiate the LCA, the user clicks the “Balance” button. *eLCAP* will perform the LCA by looping over all the events (Construction Stage and Use Stage) in the life cycle, building an LCA model for each, balancing, and then performing the Life Cycle Impact Assessment (LCIA).

The “progress” message area immediately below the buttons on the page provides feedback to the user during the LCA. There are messages for each event. Construction-type events typically take more time to execute so the messages are easier to read; Use Stage events happen very quickly so the messages pass by quickly. For longer-executing events, there will be numbers visible in the message area as *eLCAP* sends second counts to the browser during the LCA.

Figure 4.15: Analyze & Results page.

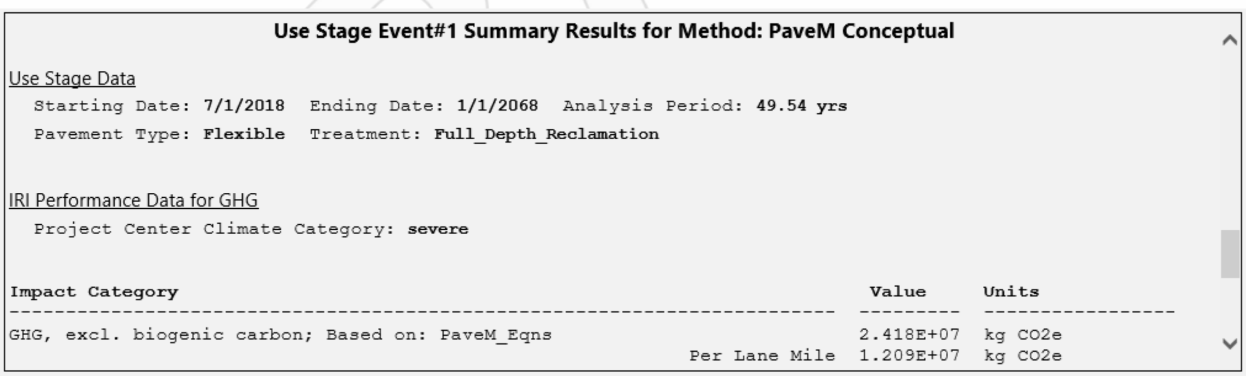


Figure 4.16: Analyze & Results page showing Use Stage event summary results.

The last messages shown for an LCA are "Report Generation Started" and "Report Generation Done." When these appear the "Get Results" button is enabled and generated report files can be downloaded.

The LCA may be stopped before it is complete by clicking the "Cancel" button.

4.7.1 Results Generation

A set of reports is generated automatically for each construction event–Use Stage pair of events. Detailed/debug level reports are generated as well as a standard *Excel* report. The list of reports for a single construction event and a single Use Stage life cycle event is shown in Figure 4.17. The *Excel* file is the standard report and all others are detailed/debug level reports.

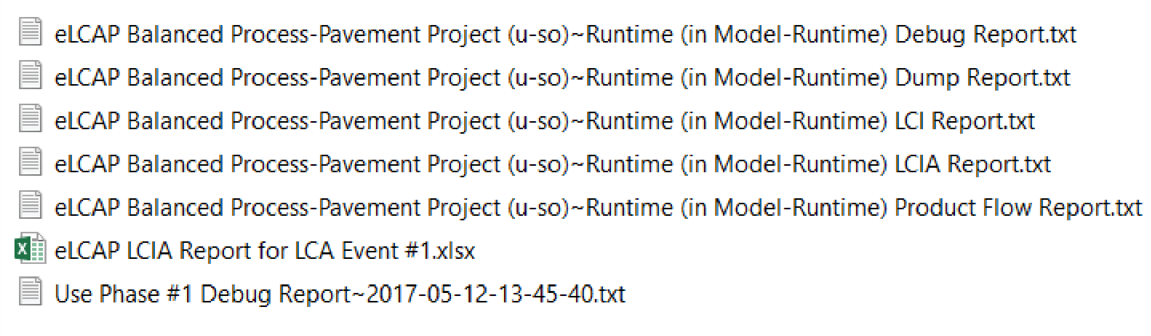


Figure 4.17: Generated reports ready for download.

4.7.1.1 Detailed/Debug Level Reports

Figure 4.17 shows the reports generated by *eLCAP* for a single construction event–Use Stage pair of events. The first five reports shown in the figure are detailed/debug level construction-type event reports. The first one listed is the most comprehensive of the reports: it gives input/output flows for every process in the Pavement Project Model (there can be hundreds of them) with the following data for each flow:

1. Flow name
2. Amount of flow "as collected"
3. Amount of the flow normalized to produce a unit value for the output product flow
4. Amount of the flow scaled as required by the Pavement Project Process to produce a single pavement project

Since the process of balancing starts at the Pavement Project Process and "climbs" upstream for each input flow into that process, the Pavement Project Process is the last one to be completed, and thus it appears at the end of the report. Figure 4.18 shows the sample results for that process.

The figure shows that there were 108 processes in the LCA model and that the Pavement Project Model has a single input flow (907.185 kg of HMA, which is a user input value of 1 ton of HMA) and one output flow (a pavement project).

Since the user specified 1 ton of HMA for the example pavement project, all flows in the HMA Process upstream from the Pavement Project Process are scaled by 907.185. The scaling is performed, recursively, upstream for each model attached to each input flow in each process.

```
Process (108) Information for: Pavement Project (u-so)~Runtime             This Process is FIXED;   Model Scale Factor: 1.000
     This is a Pavement Project (u-so) Process

                                  All Techsphere and Nature flows are scaled by: 1.000000e+00

                                                                                                  Normalized to #
                                  Flows                      As Collected           Normalized to *       Techsphere Flows    Nature Flows
---------------------------------------------------- ---------------------- --------------------------- --------------- -----------------
Inflows
  Hot Mix Asphalt (HMA) [UCPRC Flows]-1                  907.185 kg             907.185 kg/pcs          907.185 kg/pcs

Outflows
  Pavement Project [UCPRC Flows]*#                         1.000 pcs              1.000 pcs/pcs                                1.000 pcs/pcs
```

Figure 4.18: Debug Level Report results for the Pavement Project process.

Figure 4.19 shows some of the detailed information in the Use Stage event report.

The top section shows the three route segments for the example of DN-101-North, from PM R1.000 to PM R5.000. The second section shows some high-level Use Stage data. The third section shows route segment and lane data for the first year on the Use Stage. The information in this section is discussed in Section 2.7.2.

```
Segment List AFTER Project Limit Trimming (Traffic Volumes are for BOTH Directions)

                                    MgmtSegment List for Pavement Project Get Segments
---------------- Beg Segment Boundary ------------  ------- Segment ------  ---------------- End Segment Boundary ------------
   Event          CODM(m)      SODM        PM      Lanes   AADT  Trks Yr      Event          CODM(m)      SODM         PM
---------------  ----------  ----------  ---------  ----- ------ ----- --  ---------------  ----------  ----------  ---------
BegProj                0.0     763.637     M0.000      2   4450   507 00  LaneRange              0.0     767.317     R3.647
LaneRange              0.0     767.317     R3.647      1                     Traffic               0.0     768.308     R4.638
Traffic                0.0     768.308     R4.638      1                     EndProj               0.0     768.670     R5.000


Use Stage Data
          Starting Date: 1/1/2019    Ending Date: 7/1/2029    Analysis Period: 10.50 yrs
          Pavement Type: Flexible    Treatment: Full_Depth_Reclamation
          Project Length: 5.033 miles    Total Lane Miles: 8.713 miles    Avg # Lanes 1-Dir: 1.73

          Project Center Climate Zone: North Coast    Climate Category: severe
          Traffic Growth Rate: 0.0%


Use Stage GHG Due To IRI (Traffic Volumes Reflect Growth Rate Effects from Midpoint of Traffic Count Year to Date Shown in 'Date' Column)

Yr#    Date     Year Seg#  Seglen C WIM Ln#   DF     AADT   CarVol  TrkVol 2-Axle 3-Axle 4-Axle 5-Axle ESAL E   A     B   C   IRI       GHG
                            (mi)                                                                        (k)                    (ipm)      (kg)
--- ---------- ---- ---- ------- - --- --- ----- ------- ------- ------- ------ ------ ------ ------ ---- - ----- ---- --- ----- ------------
  1  7/1/2019  0.5    1    3.680 s  1b   1 0.190   1,152   1,104      48     17      8      0     20   24 A 157.3  3.4 1.0 159.0 7.343863E+05
                                         2 0.810   1,072     868     204     75     37      2     89  101 B 143.0  5.1 1.0 145.6 1.006404E+06
                          Segment Totals:  1.000   2,224   1,972     252     92     45      2    109                              1.740790E+06

                     2    0.991 s  1b   1 1.000   2,225   1,972     253     93     46      3    110  124 B 143.0  5.1 1.0 145.6 4.707014E+05
                          Segment Totals:  1.000   2,225   1,972     253     93     46      3    110                              4.707014E+05

                     3    0.362 s  1b   1 1.000   2,300   2,063     237     61     31     50     94  117 B 143.0  5.1 1.0 145.6 1.787601E+05
                          Segment Totals:  1.000   2,300   2,063     237     61     31     50     94                              1.787601E+05
                                                                                                                 Year Total:  2.390251E+06
                                                                                                   Year Total Per Lane Mile:  2.743316E+05
```

Figure 4.19: Debug Level Report results for the Use Stage.

4.7.1.2 Standard *Excel* Report

The following sections show some of the results for the standard *Excel* report.

Sample Project-Level bar chart. Figure 4.20 shows 6 of the 18 computed impact categories, with each chart showing results for "Material Production," "Transport," "Construction Equipment," and "Construction."

Figure 4.20: *Excel* report showing bar charts.

Sample Material-Level bar chart. Figure 4.21 shows 2 of the 18 computed impact categories, with each chart showing several different types of grouped results.

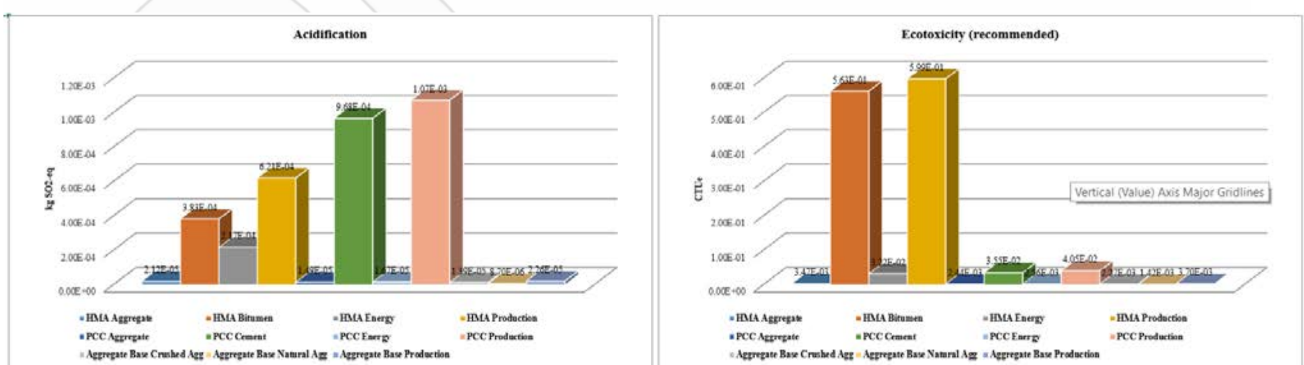


Figure 4.21: *Excel* report showing 3D bar charts.

Sample result table. Figure 4.22 shows the Project Totals table.

Project Totals

Selected Impact Categories and Inventories	Material Production	Material Production as % of Construction	Transport	Transport as % of Construction	Construction Equipment	Construction Equipment as % of Construction
Acidification	1.7178E-03	93.89%	1.1011E-04	6.02%	1.7637E-06	0.10%
Ecotoxicity (recommended)	6.4297E-01	92.66%	5.0638E-02	7.30%	3.2940E-04	0.05%
Eutrophication	5.5258E-05	89.74%	6.2132E-06	10.09%	1.0523E-07	0.17%
Global Warming Air, excl. biogenic carbon	3.3580E-01	94.66%	1.8821E-02	5.31%	1.2715E-04	0.04%
Global Warming Air, incl. biogenic carbon	3.3581E-01	94.49%	1.9461E-02	5.48%	1.2715E-04	0.04%
Human Health Particulate Air	1.4940E-04	96.07%	6.0149E-06	3.87%	9.9763E-08	0.06%
Human toxity, cancer (recommended)	2.2114E-10	97.77%	5.0047E-12	2.21%	4.3398E-14	0.02%
Human toxity, non-cancer (recommended)	5.2534E-08	94.94%	2.7842E-09	5.03%	1.8156E-11	0.03%
Ozone Depletion	1.1043E-09	99.93%	7.9885E-13	0.07%	5.1961E-15	0.00%
Resources, Fossil fuels	5.5400E-01	93.65%	3.7342E-02	6.31%	2.4289E-04	0.04%
Smog Air	2.4085E-02	88.74%	3.0006E-03	11.06%	5.6148E-05	0.21%
Primary Energy Demand used as raw materials (Feedstock Energy)	2.4120E+00	100.00%	0.0000E+00	0.00%	0.0000E+00	0.00%
Primary energy demand from ren. and non ren. resources (gross cal. value)	5.1562E+00	94.67%	2.8825E-01	5.29%	1.8749E-03	0.03%
Primary energy demand from ren. and non ren. resources (net cal. value)	4.8278E+00	94.68%	2.6936E-01	5.28%	1.7520E-03	0.03%
Primary energy from non renewable resources (gross cal. value)	5.0339E+00	94.55%	2.8825E-01	5.41%	1.8749E-03	0.04%
Primary energy from non renewable resources (net cal. value)	4.7055E+00	94.55%	2.6936E-01	5.41%	1.7520E-03	0.04%
Primary energy from renewable resources (gross cal. value)	1.2232E-01	100.00%	0.0000E+00	0.00%	0.0000E+00	0.00%
Primary energy from renewable resources (net cal. value)	1.2232E-01	100.00%	0.0000E+00	0.00%	0.0000E+00	0.00%

Figure 4.22: *Excel* report showing data table.

5 ARCHITECTURE

eLCAP implements the well-known three-layer architecture to support separation of functionality and to promote code reuse:

- User Interface, UI
- Business Layer, BLL
- Data Access Layer, DAL

Any given lower-level layer (e.g., Data Access) does not know anything about the layers above it (e.g., Business and the UI). In addition, any given upper-level layer (e.g., UI) communicates with lower-level layers (e.g., Business) using an API (data in a layer is private to that layer and API functions provide access to its data and operations on its data). And in general, upper-level layers communicate only with the layer immediately below it, but there are exceptions.

The UI is designed to be as “thin” as possible with all business-type activities handled in the BLL. Error checks are made in the UI to provide the best possible feedback for the user, but error checks are also made in the other layers. Implementing a user-friendly UI tends to be a time-consuming activity since it is crucial to the acceptance and overall use of the application.

The BLL consists of a large set of classes. Currently, there are around 800 C# classes/interfaces in the BLL. The BLL lives in a dynamic link library (DLL) named “Utilities.” The BLL models the business domain of the UCPRC and is organized as shown in Figure 5.1. Almost all classes in the BLL inherit from a base class, *UCPRCBase*, which contains member data needed by all classes in the BLL and virtual functions implemented by all classes.

The BLL is general (i.e., not specific to LCA) so it is used in applications other than *eLCAP*. The *eLCAP*-specific part of the architecture is the UI; the BLL and the DAL can be used by any application. The size of the Utilities DLL in debug configuration is around 3 MB.

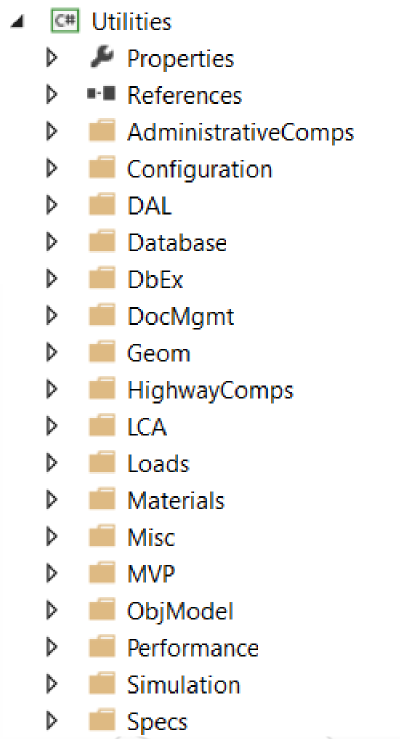


Figure 5.1: Groups of classes in the Utilities DLL.

The BLL accesses the DAL with high-level, domain object-level data requests, and is agnostic about the specific data storage type and location. The details of the specific data storage and location of that storage is specified in the configuration file, *web.config*, and the lowest-level set of DAL functions. The architecture of the DAL implements the Factor Pattern and Data Provider Pattern to make it relatively straightforward to change data storage types in the future and to allow every specific data object (e.g., a table) to be stored in separate data stores and locations: all data tables need not be stored in the same physical database.

The basics of the Factory Pattern and Data Provider Pattern used for the DAL is that data access is provided to upper-level classes via an “abstract” class. This class defines abstract functions (usually to get or update database data) that must be implemented by a concrete implementation, such as a class containing functions to access an SQL Server. The specific concrete class to use at runtime for a particular table of data is set in the configuration file or it can be set by some other means. The Factory Pattern implemented in the DAL is shown in Figure 5.2. The Provider Pattern is similar but slightly simpler.

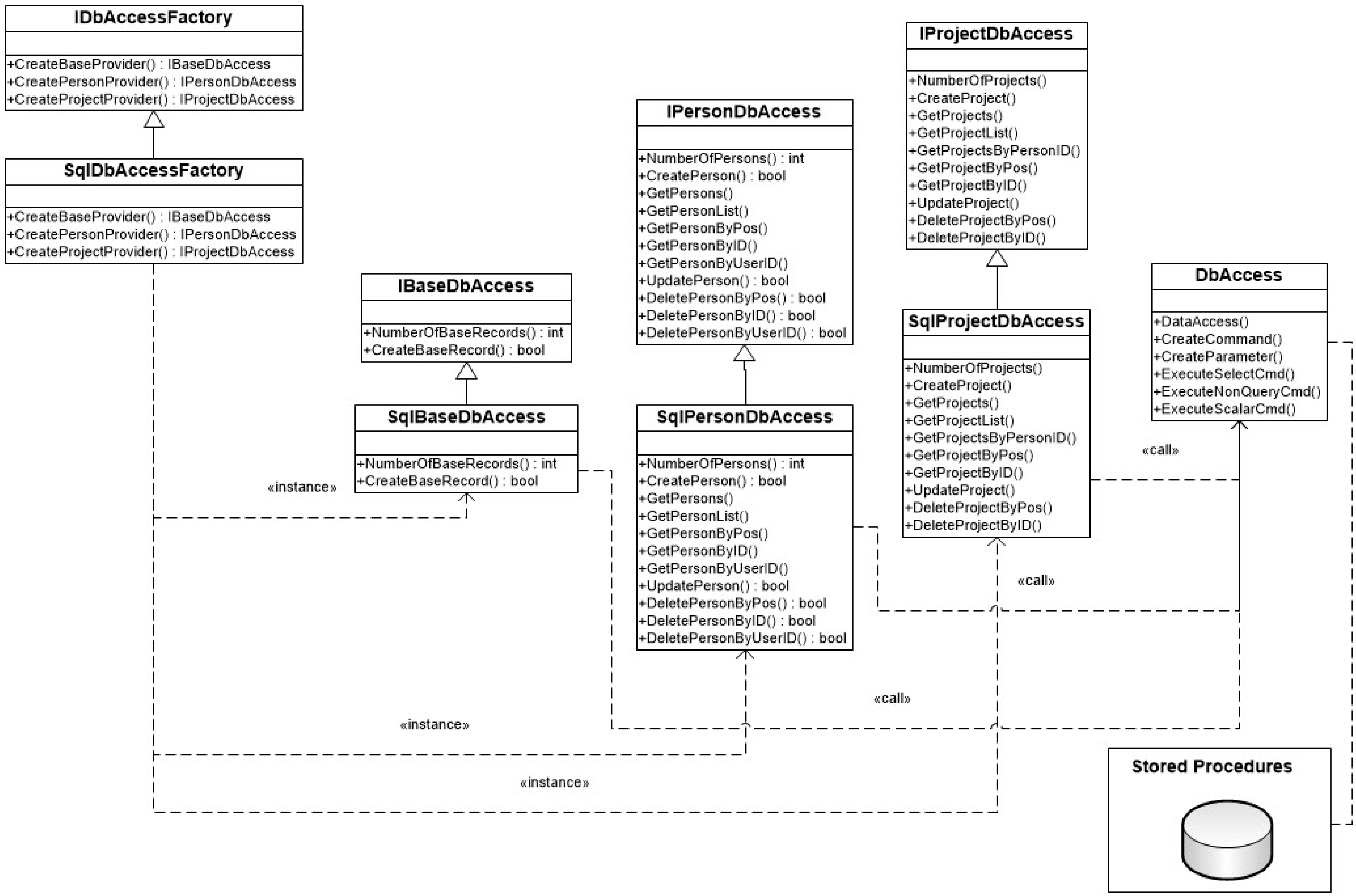


Figure 5.2: Factory Pattern implemented in DAL.

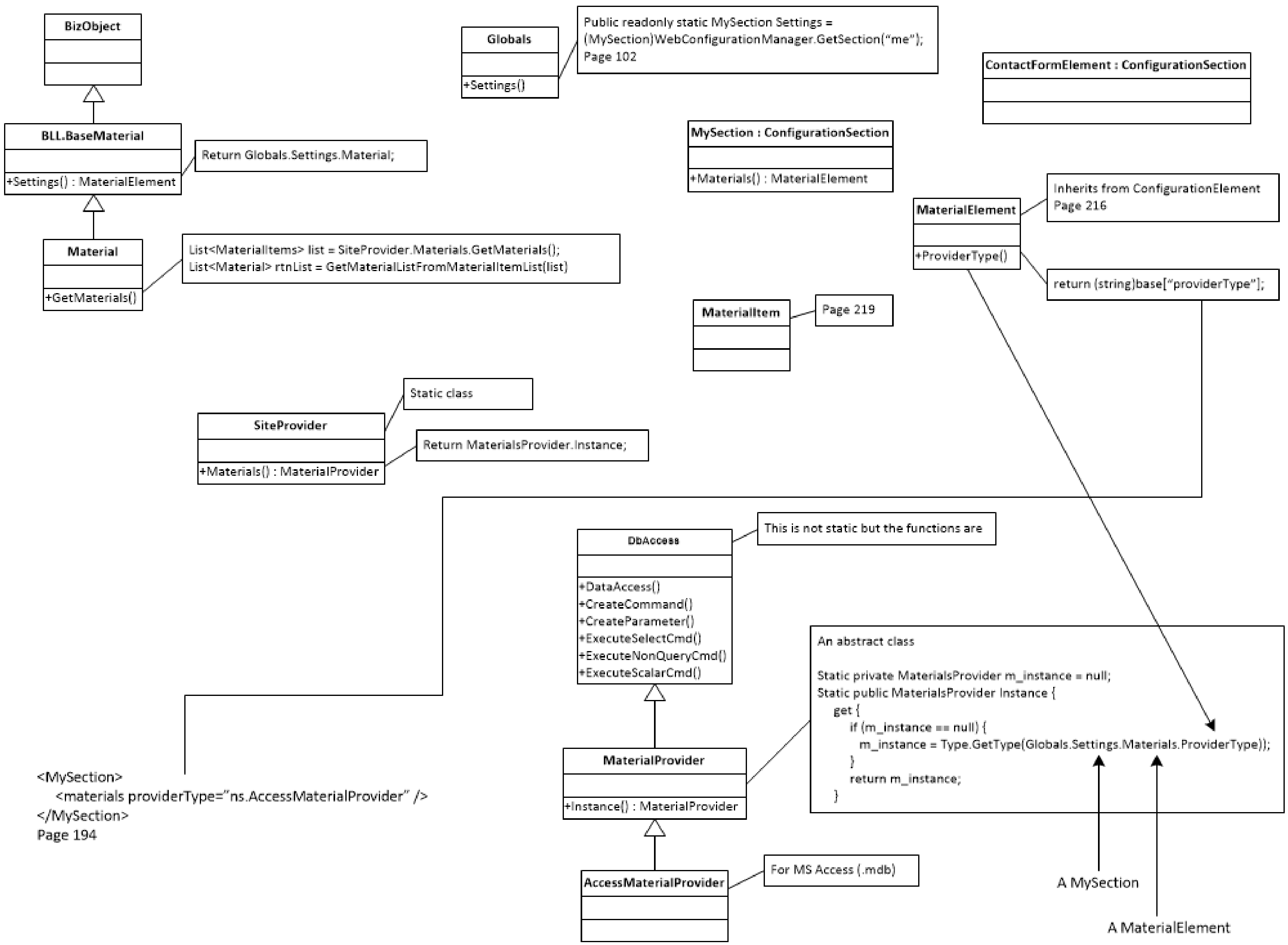


Figure 5.3: Provider Pattern implemented in DAL.

eLCAP is an ASP.NET (web) C# application so its internal organization is influenced by ASP.NET, and it also makes use of many services of the .NET runtime environment. Currently, *eLCAP* is built to the 4.5.1 version of .NET runtime. Development is done using *Visual Studio* (2013).

eLCAP also makes use of several third-party libraries (see Section 5.7). These libraries are managed by the *Visual Studio* included *NuGet* package manager.

5.1 ASP.NET Concepts

The UI in *eLCAP* is implemented using ASP.NET, a mature Microsoft web technology initially released in 2002 as *Active Server Pages* (ASP) that later turned into ASP.NET. It is a server-side web framework built on the .NET Common Language Runtime (CLR) and currently supports many different programming models, such as MVC. ASP.NET can be used with any programming language supported by the CLR; the UI in *eLCAP* uses C#. The CLR consists of many thousands of classes and provides an extensive array of services to the application developer.

A web application is, by HTTP design, a stateless machine: each request to a web server results in the web server constructing completely new versions of all controls and the page itself. The web application does not “remember” what happened on an earlier request. This is very different from a desktop application, in which state is maintained during a user’s interaction with it. ASP.NET provides several techniques to maintain state between multiple page requests in order to provide needed UI functionality. *eLCAP* uses the ASP.NET *Session State* facility to maintain a finite amount of program state. In general, one tries to minimize the amount of Session State since it consumes valuable web resources, but maintaining state is necessary in any user-friendly UI.

Client-side programming can easily be done using JavaScript downloaded to the browser at runtime. *eLCAP* makes use of this in various ways to extend the functionality of the server controls and to provide a more responsive UI.

5.1.1 HTML (aspx, server controls, CSS), C# Code behind and Event and Error Handling

The basic pattern in ASP.NET is that visual controls such as data fields, radio buttons, check boxes, and tables (grids) are implemented using ASP.NET “server”-side controls located in *.aspx files (very similar to HTML files and HTML controls) and user events (for example, clicking a check box or entering data in a field are “handled” or trapped in C# “code behind” files). The C# functions that respond to user interactions provide the main UI functionality and the interaction with the BLL to set and get data and perform any needed UI operations using the facilities of the BLL. The ASP.NET server controls, when loaded by the ASP.NET web server (IIS), emit standard HTML and any necessary JavaScript to the web browser.

The visual aspects of the server and HTML controls and pages are handled using standard CSS. ASP.NET allows for formatting of controls within the server-side controls themselves in the aspx file but it is performed in *eLCAP* via CSS for greater flexibility and to follow web standards.

Error trapping and handling is a complex and detailed activity. The first layer of error trapping and handling is done using ASP.NET server Validation controls to perform range checking, data existence, etc. These Validation controls are implemented in JavaScript that is generated by the ASP.NET engine. The second layer of error trapping and handling is done in the C# code behind, and messages are presented to the user as necessary by the code behind code. The third layer of error trapping is done in the BLL; standard C# exceptions are "thrown" by BLL functions and "caught" by the UI code with subsequent issuing of user messages.

5.1.2 Master and Content Pages

The UI in *eLCAP* uses a facility in ASP.NET called "Master" and "Content" pages. A Master aspx page is where common page look-and-feel and functionality are placed while specific page content is placed in separate Content pages. A Content page inherits the controls and code from a Master page. This method minimizes coding by having common things in one place and provides a simple mechanism to have a consistent look-and-feel across all pages in a web application.

5.1.3 Web.config

A critical file in the world of ASP.NET is *web.config.* This is an XML-formatted file that contains many sections, and it is where users are able to control many aspects of the application without having to rebuild and redeploy the application. ASP.NET detects that this file has been edited and reloads it automatically. Since web.config is an XML file, control of the application is done using "name-value" pairs (e.g., "<add key="my important key" value="key value" />).

The main sections of web.config used in *eLCAP* are:

- *ConnectionStrings:* this section is used to specify connection parameters to database sources. Items such as the server url, type of security used by the database server, the name of the database, user log-in credentials and specific data provider (SQL, OLEDB, etc.) used to access the data are defined.

- *AppSettings:* this section is used to define application data that can easily be changed by simply editing web.config on the web server. Items such as directory locations of runtimes files, email settings, and fully qualified names of classes used by the DAL to access specific tables of data are defined.
- *Authentication, RoleManager, Membership and Profile:* these sections are used to control user authentication and authorization and to keep track of user profiles (e.g., which Caltrans district a user is in).
- *Page Protection:* this section allows specific pages of the application to be protected so that only specific authorized users or specific groups of users can access them.

5.2 Authentication, Authorization, Profiles and the Database

The design and implementation of *eLCAP* included user access controls via standard log-in procedures. ASP.NET has built-in support for user authentication and authorization and group membership. An SQL Server database is provided by ASP.NET, called *aspnetdb* which is used to store all user access/log-in data. The database schema is shown in Figure 6.1. ASP.NET provides customizable log-in controls (see Section 4.1) for registration, log in, password recovery, etc.

eLCAP adds some custom fields to the Registration control to gather data that is not part of the built-in control. Data items such as First Name, Last Name, and Caltrans District have been added and they are stored a user profile in the aspnetdb database. In addition, the most recently used project and project trial are stored in the user profile so *eLCAP* can open up the project trial used last upon starting a new *eLCAP* session.

5.3 Automatic Email Generation

eLCAP sends email to users and the system administrator for a variety of reasons. When a person registers to become a member of *eLCAP*, an email is sent to the system administrator so he/she can verify the person, and then authorize and put the person in the appropriate user group.

In addition, if the application crashes during use, an email containing a variety of debug information is sent to the system administrator (for later debugging) and a "user-friendly" page is shown to the user.

5.4 Units

Unit conversion in engineering applications is an ongoing issue but one that needs to be addressed at the very beginning of design and development. *eLCAP* uses the services of a unit conversion subsystem called *EngrUnits* that converts between many different types of units. Currently, there are eight different quantities: area, energy, force, length, mass, time, and volume in *EngrUnits*. Each quantity contains many variations of the units. For example, length has 20 different units (e.g., kilometer, meter, mile, league, yard, chain, twip, etc.).

To get a conversion factor to convert Miles to Feet:

```
milesToFeet = Length.Factor(Length.Type.Mile, Length.Type.Foot);
```
(1)

5.4.1 Base and Client

The *EngrUnits* subsystem forms the basis of a slightly higher, application specific, and more convenient method of performing unit conversion. *eLCAP* sets up a base set of units forming a consistent set of units for all internal computations. This set of base units is known only to *eLCAP*. All data that is given/specified to the internal data structure in *eLCAP* is converted to base units using the services of the *EngrUnits* subsystem. The BLL has a class, called *Units*, that has a set of units for the base and a set of units for the client (the UI); *eLCAP* establishes the set of base units and set of client units during startup.

Whenever the UI needs to specify data to its internal project data structure or BLL data structures, it gets a conversion factor. For example, to get a conversion factor to convert client data (mile) to base data, the following is used:

```
clientToBase = Length.Factor(Length.Type.Mile, Units.BaseLengthUnits);
```
(2)

The reverse is done when data are extracted from *eLCAP*'s internal data structure and displayed in the UI.

For cases where a particular item in the UI is not part of the client set of base units, the above approach will not work and the approach in (1) above is used. For example, if a client unit for length is *meters* but a particular data item exposed to the user is in *kilometers*, then (1) is used.

5.4.2 Per Data Item

eLCAP is an LCA tool and works with flows of materials and chemicals, and so it must deal with a wide range of different units in the UI (and also the actual database of processes and flows) since quantities of materials (e.g., HMA) might be known to the user in tons or tonnes, kilograms (kg) or grams (g), pounds, or something else. Therefore, *eLCAP* allows the user to select, from a list of units, a specific unit for a specific data item. For example, when defining a construction-type event, a user may know the amount of AB in kilograms but the amount of HMA in tons.

To avoid forcing a user to convert one or both to whatever the UI wants for quantity, *eLCAP* allows the user to specify AB in kilograms and HMA in tons. To support this, *eLCAP* maintains a unit specification for the quantity data item (and others), and during the balancing process (see Section 2.1) all flow quantities are converted to the same base unit using (2) above.

5.4.3 UI and C# Code

ASP.NET provides several methods of making the localization process as simple as possible (i.e., to minimize the time for it to happen).

For UI controls, the following is used:

```
Text="<%$ Resources:GlobalResources, EndOfLifeLabelPartB_G %>">
```

ASP.NET will look in a string resource file named *GlobalResources.??.resx*, where "??" is the two-letter ISO language code (e.g., "en," "es," "fr," etc. for the string labeled "`EndOfLifeLabelPartB_G`") and assign that string to the Text of the control.

For getting a string in code, the following is used:

```
Text = Resources.GlobalResources.EndOfLifeLabelPartB_G
```

5.5 User Real-Time Feedback: SignalR

eLCAP is web application, and as such, it is constrained by web protocols. One constraint is that a single *response* is sent by a server when it receives a *request* from a client browser. Clicking a button is a request; sending a message from the server to show the user is a response. This model works perfectly well for most web applications. However, it does not when the request results in a long-running execution of the application. Performing an LCA is a long-running activity and user feedback is necessary for a user-friendly UI.

Sending user feedback during a long-running web application requires the use of real-time web technology. The specific real-time technology used by *eLCAP* is called *SignalR*. It is basically a point-to-point, bidirectional connection technology, similar to chat applications, that allows the server to send content to the client browser as it happens, in real-time, independent of a web request.

SignalR uses WebSockets, when available, as the communication technology, and older protocols as necessary. SignalR provides a very simple, high-level API for doing server-to-client RPC (Remote Procedure Calls, i.e., call JavaScript functions in the browsers from server-side .NET code) in the ASP.NET application.

Clicking the "Balance" button on the Analyze & Results page results in calling a function, located in a special class in *eLCAP*, which deals with real-time, client-to-server and server-to-client communication. All messaging to the client browser during the LCA is done using SignalR.

5.6 Main Classes

As mentioned earlier, the BLL contains over 800 classes/interfaces and it is neither practical nor instructive to discuss all of them here. This section briefly touches on the more important classes.

5.6.1 Base Class

Most of the classes in the BLL derive from the base class *UCPRCBase*. This class is shown in Figure 5.4. It is very convenient to have classes derive from a base class for a variety of reasons.

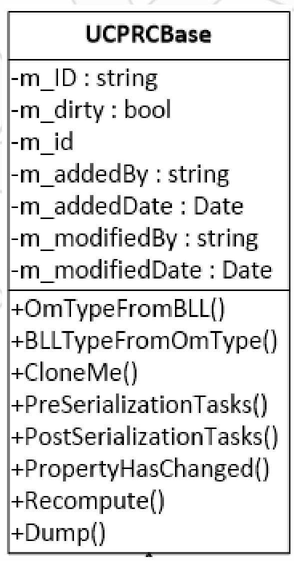


Figure 5.4: The Base Class, *UCPRCBase.*

5.6.2 Life Cycle

The pavement project life cycle is modeled by two classes, *LcaLifeCycle* and *LcaEvent*. An *LcaEvent* object is added to the list of *LcaEvents* in *LcaLifeCycle* for each construction-type event defined by the user, and then *eLCAP* adds another one after it to represent the Use Stage. The *m_pavementModel* data member in *LcaEvent* either contains a *PavementProjectModel* or a *PavementUseModel* object.

Stage 4

The *BuildLcaModel*() function in *LcaEvent* translates either *PavementProjectModel* object into a *PavementLca* object or a *PavementUseModel* object into a *PavementUseLca* object. The "Lca"-named objects are the objects that know how to do an LCA analysis. The *PavementLca* object does the balancing of flows for the generated process based model (see Section 2.7.1); the virtual *Balance*() function for *PavementUseLca* does nothing since the Use Stage is based on performance models (see Section 2.7.2).

When a user requests that *eLCAP* perform an analysis, *eLCAP* iterates through the *LcaEvent* list in *LcaLifeCycle* calling *BuildLcaModel*(), *Balance*() and *DoLcia*().

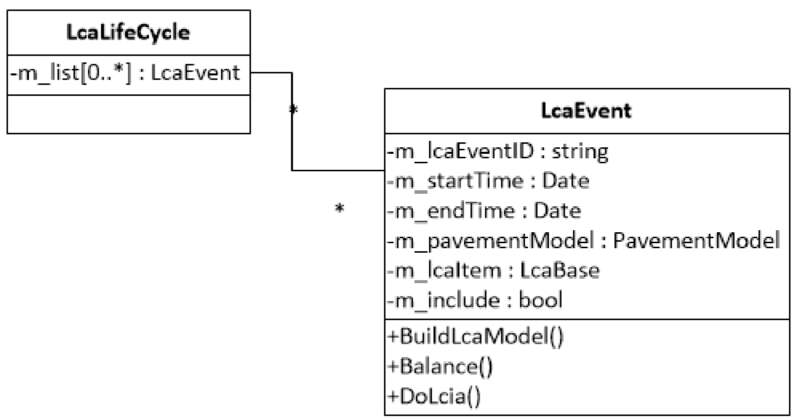


Figure 5.5: Life cycle classes.

5.6.3 Important LCA Classes

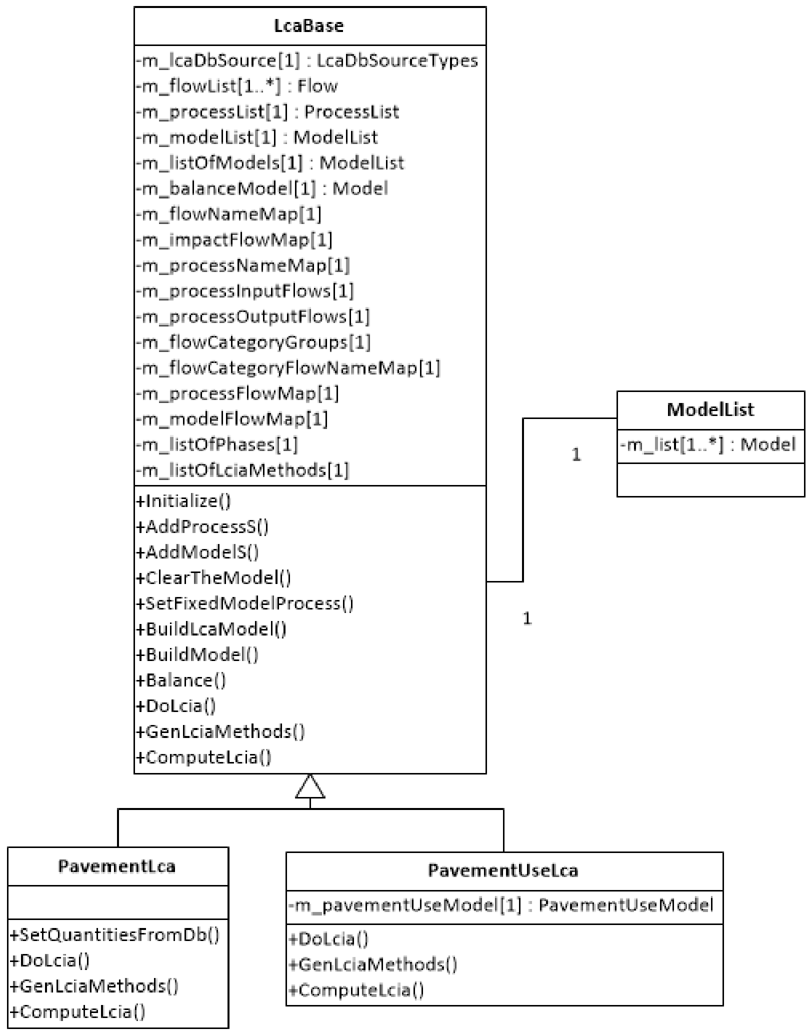


Figure 5.6: *LcaBase, PavementLca,* and *PavementUseLca* classes.

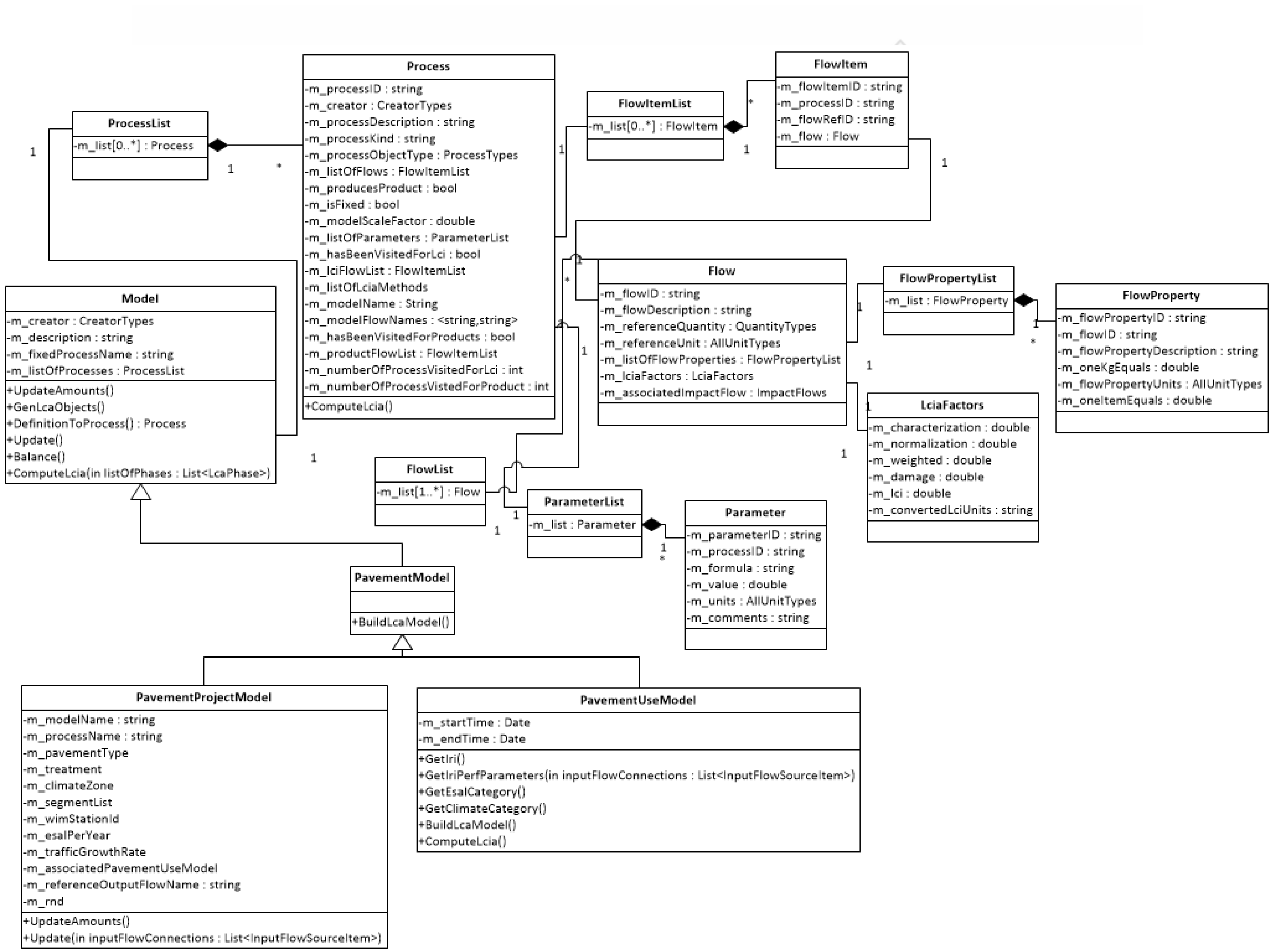


Figure 5.7: Model and process-related classes.

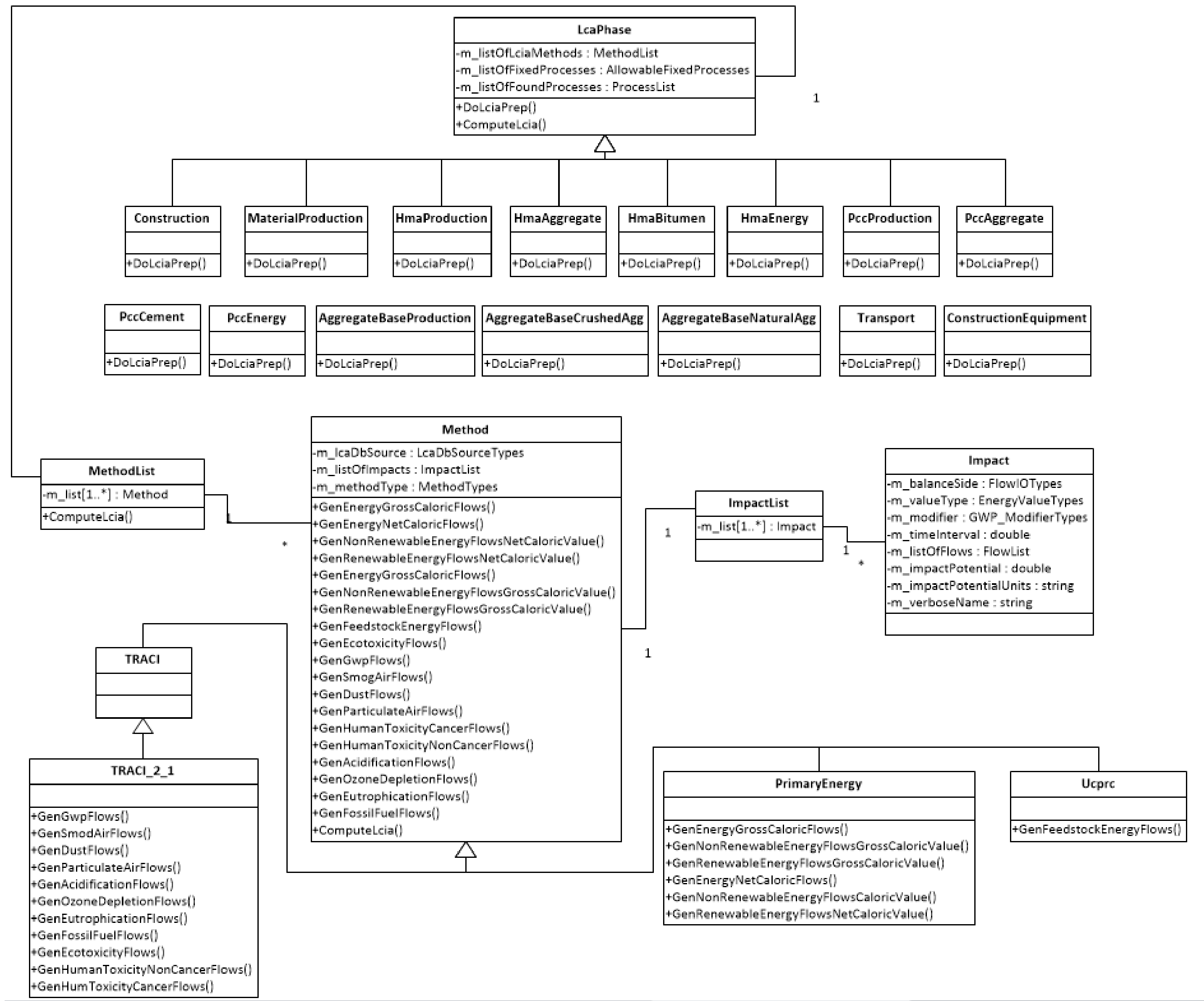


Figure 5.8: LCIA assessment-related classes.

5.7 Third-Party Libraries

eLCAP uses a few third-party libraries managed by *Visual Studio*'s *NuGet* Package Manager:

- *AjaxControlToolKit:* The ASP.NET AJAX Control Toolkit is an open-source project built on top of the Microsoft ASP.NET AJAX framework. It is a joint effort between Microsoft and the ASP.NET AJAX community that provides a powerful infrastructure to write reusable, customizable and extensible ASP.NET AJAX extenders and controls, as well as a rich array of controls that can be used out of the box to create an interactive Web experience.
- *SharpZipLib:* zip-file services
- *MathNet.Numerics:* Math.NET Numerics provides methods and algorithms for numerical computations in science, engineering, and everyday use. Covered topics include special functions, linear algebra, probability models, random numbers, interpolation, integration, regression, optimization problems, and more.
- *Newtonsoft.json:* json file serialization and deserialization support
- *Spire.xls: Excel* file generation services
- *Mathos.Parser.MathParser:* expression parsing services
- *Select.HtmlToPdf:* HTML-to-PDF file services

5.8 Application/Session Startup

eLCAP performs several activities when a user starts a new web browser session:

- UI control Session State variables are initialized.
- UI language Session State variables are initialized.
- Base and Client units are set.
- Some main classes are initialized.
- Permissions for user groups (Roles) are set.
- Directory locations for runtime files are set.
- Support files are read into memory using .NET tasks.
    - Caltrans Highway Log
    - Caltrans Route Direction

5.9 Software Development

5.9.1 The Environment

All development for *eLCAP* is done using *Visual Studio* 2013 and targeting the version 4.5.1 of the .NET runtime.

5.9.2 NuGet

Third-party libraries are managed using *NuGet* provided by *Visual Studio.*

5.9.3 Source Control

All source files are managed by the source control system “Subversion” (svn) and using the client visual tool called *TortoiseSVN.*

6 DATA ACCESS

eLCAP makes use of data from several sources. It makes three formal database connections: *SQL Server*, *MS Access*, and the log-in database, and it loads its LCA XML database file directly into memory for performance reasons. As discussed in Chapter 5, *eLCAP* uses either a Factory Pattern (for the XML data) or the Provider Pattern (for all other data except Log in) to access the data. Access to log-in data are provided by a .NET API.

6.1 Database Connections

Database connection specifications are defined in the ASP.NET configuration file, *web.config*.

6.1.1 MS SQL Server

Project, Project Trial, User-Defined Processes, and IRI Performance Model parameters data are stored in a MS *SQL Server* database having the schema shown in Figure 6.1. *eLCAP* connects to *SQL Server* using the connection string shown in Figure 6.2.

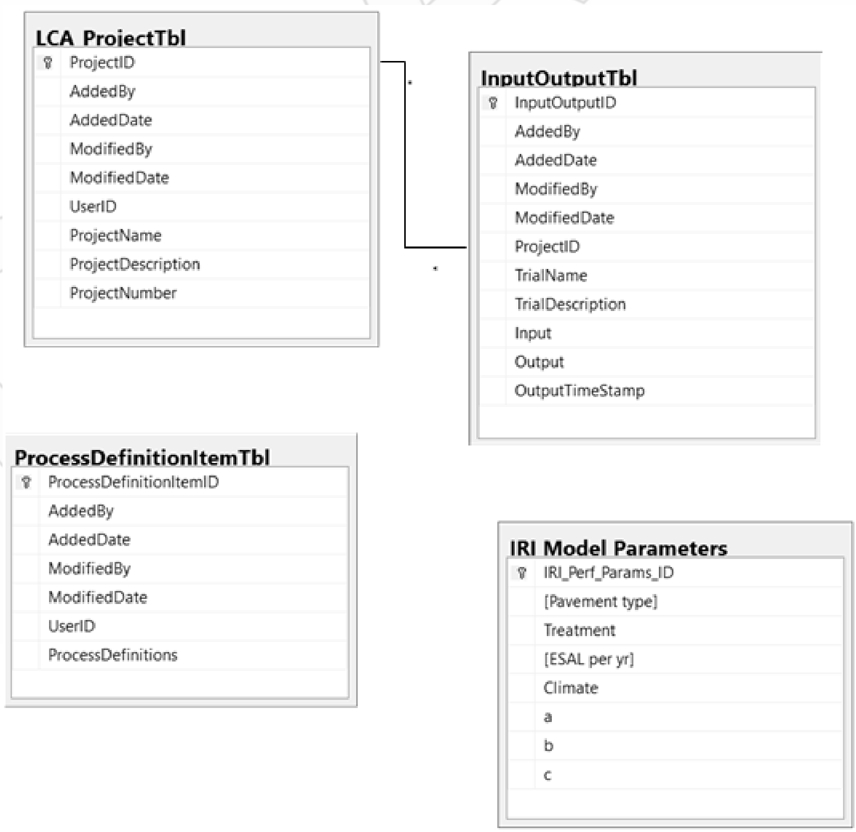


Figure 6.1: *eLCAP* database schema.

```
<add    name="eLCAP"
        connectionString="Server=dev.ucprc.ucdavis.edu;
        Integrated Security=false;
        Database=eLCAP_1_0;
        User ID=An eLCAP User;
        Password=myPwd"
        providerName="System.Data.SqlClient" />
```

Figure 6.2: SQL Server project data connection string.

All project trial data are managed by a class that is serialized to a json string and stored in the table named “InputOutputTbl” in the field named “Input.” The same is done for output but it is placed in the field named “Output.” The json string is deserialized back into the class when the data are loaded from the database.

User-defined process data are stored, as json strings, in the table named “ProcessDefinitionsItemTbl” in the field named “ProcessDefinitions.”

IRI performance model parameter data are stored in the table named “IRI Model Parameters.” The data for this table originates from a CSV file which is imported into the *SQL Server* table.

Log-in data are also stored in a *SQL Server* database; its schema is shown in Figure 6.3 and the connection string is shown in Figure 6.4. The basic log-in database is created by running a command outside of *Visual Studio.*

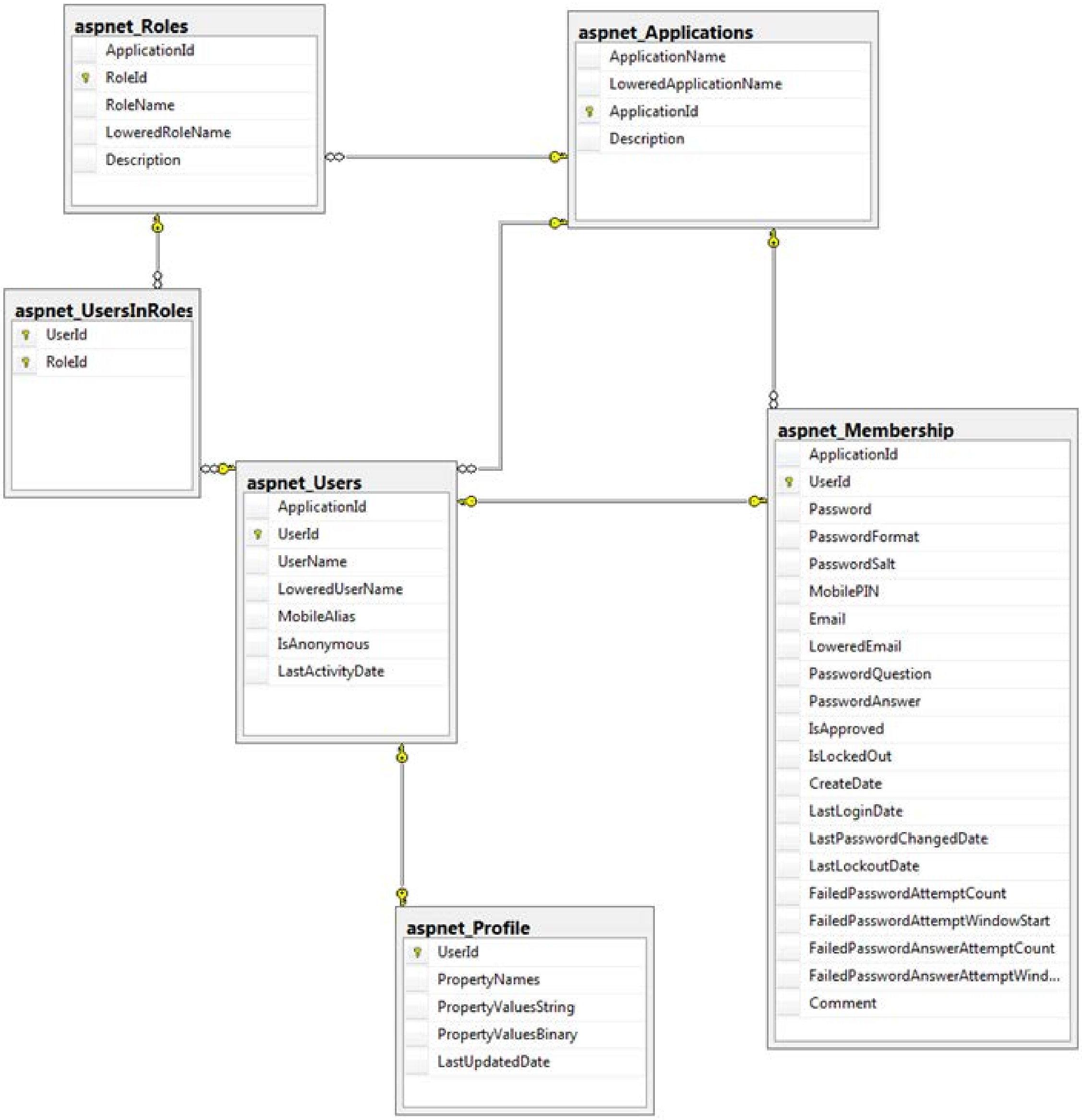


Figure 6.3: ASP.NET user authentication, authorization, and profile database schema.

```
<add    name="LocalSqlServer"
        connectionString="Server=dev.ucprc.ucdavis.edu;
        Integrated Security=false;
        Database=aspnetdb;
        User ID=Login User;
        Password=myPwd"
        providerName="System.Data.SqlClient" />
```

Figure 6.4: *SQL Server* log-in connection string.

6.1.2 MS Access

Traffic, WIM, and climate data are stored in a MS *Access* database having the schema shown in Figure 6.5, Figure 6.6, Figure 6.7, and Figure 6.8, with the connection string shown in Figure 6.9.

Field Name	Data Type
Route	Number
Route Suffix	Short Text
District	Number
County	Short Text
Postmile Prefix	Short Text
Postmile	Number
Leg	Short Text
AADT Total	Number
Total Trucks	Number
Total Trucks %	Number
2 axle volume	Number
2 axle percent	Number
3 axle volume	Number
3 axle percent	Number
4 axle volume	Number
4 axle percent	Number
5 axle volume	Number
5 axle percent	Number
Description	Short Text
Year	Short Text
Verify/Estimate	Short Text

Figure 6.5: MS *Access* traffic database schema.

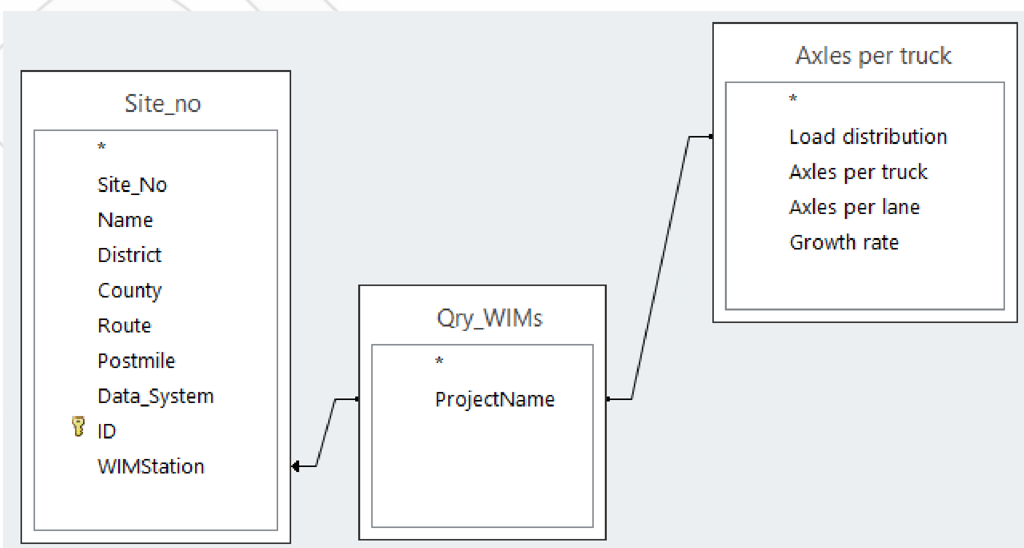


Figure 6.6: MS *Access* WIM database schema.

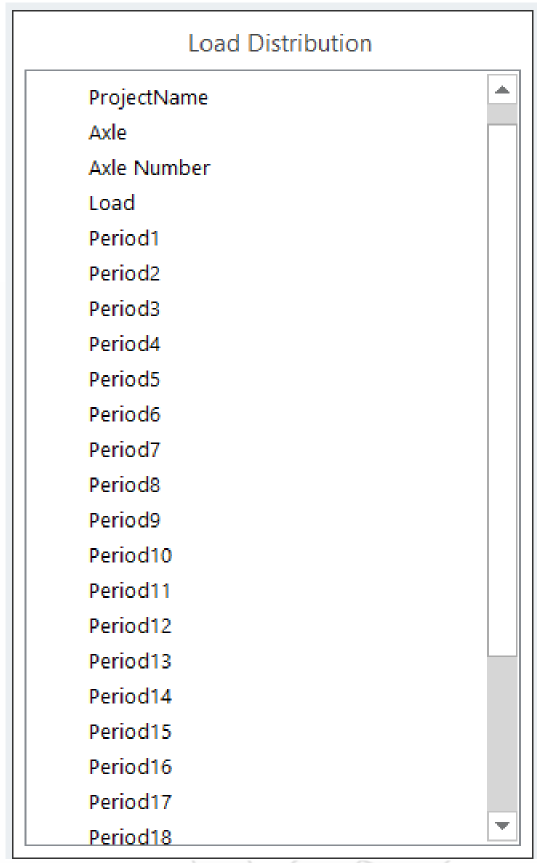


Figure 6.7: MS *Access* axle load spectra (24 hours) database schema.

Field Name	Data Type
District	Number
County	Short Text
Route	Number
Postmile	Short Text
Climate	Short Text

Figure 6.8: MS *Access* climate database schema.

```
<add    name="CalMEConnectionString"
        connectionString="Provider=Microsoft.Jet.OLEDB.4.0;
        Data Source=|DataDirectory|\CalME.mdb"
        providerName="System.Data.OleDb" />
```

Figure 6.9: MS *Access* connection string.

6.2 XML Database File

eLCAP's LCA database is an XML file that is read into memory when the application starts up. This is done for performance reasons since access to it occurs frequently. The file is constructed as discussed in Section 2.5. The schema for the file, and the in-memory version of it, is shown in Figure 6.10. Data are accessed using the DAL, as usual, for any data source used by *eLCAP*.

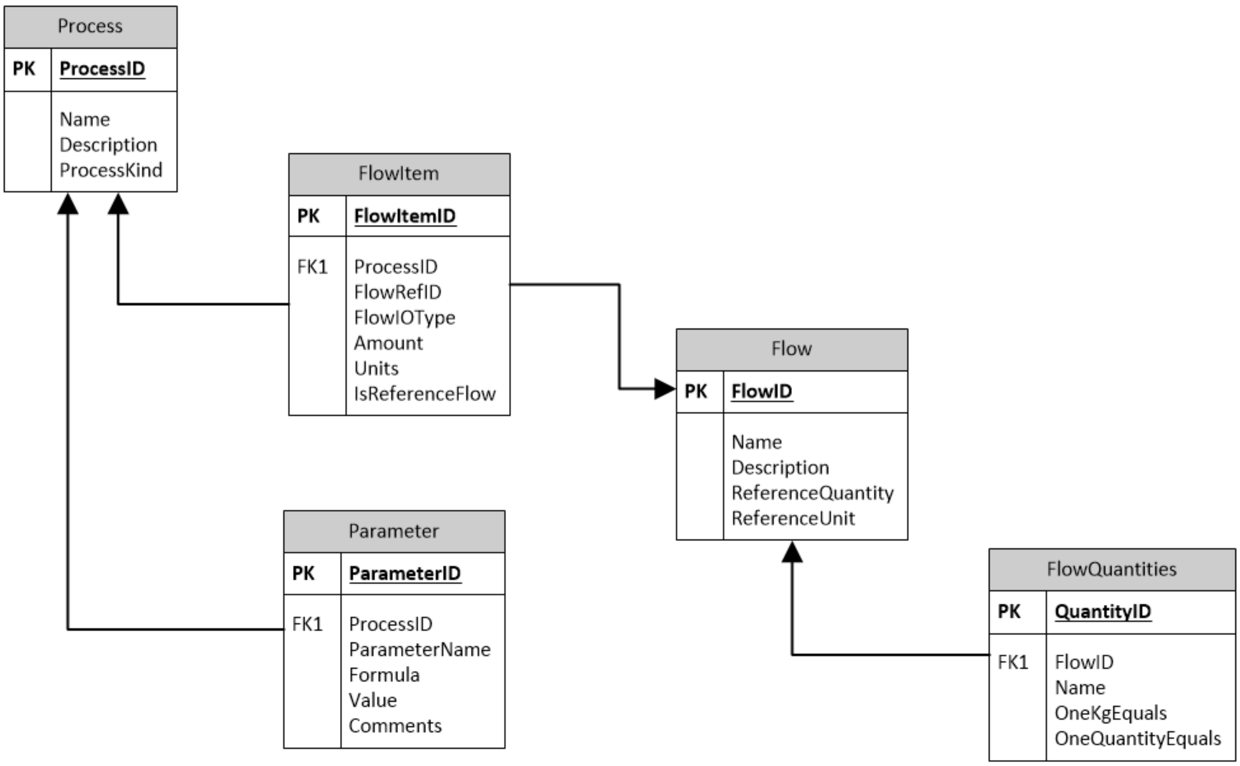


Figure 6.10: XML file schema.

REFERENCES

1. *GaBi* software. thinkstep.com, Hauptstraße 111-113, 70771 Leinfelden-Echterdingen, Germany *https://www.thinkstep.com/software/gabi-lca*
2. Lu, Q. 2008. Estimation of Truck Traffic Inputs Based on Weigh-in-Motion Data in California. University of California Pavement Research Center. UCPRC-TM-2008-08 *(in progress)*

APPENDIX A: PROCESS MODELS

The following figures show the *process models* used in *eLCAP*. A process model contains a process that produces a product, such as *HMA* or a *Pavement Project*, and other "agg"-type processes. The main product-producing process has a series of input flows that connect to either the "agg"-type processes in the model or to other models.

The figures in this appendix usually contain more than one process model. This is done to provide information about the source of the input flows for the process model in the figure.

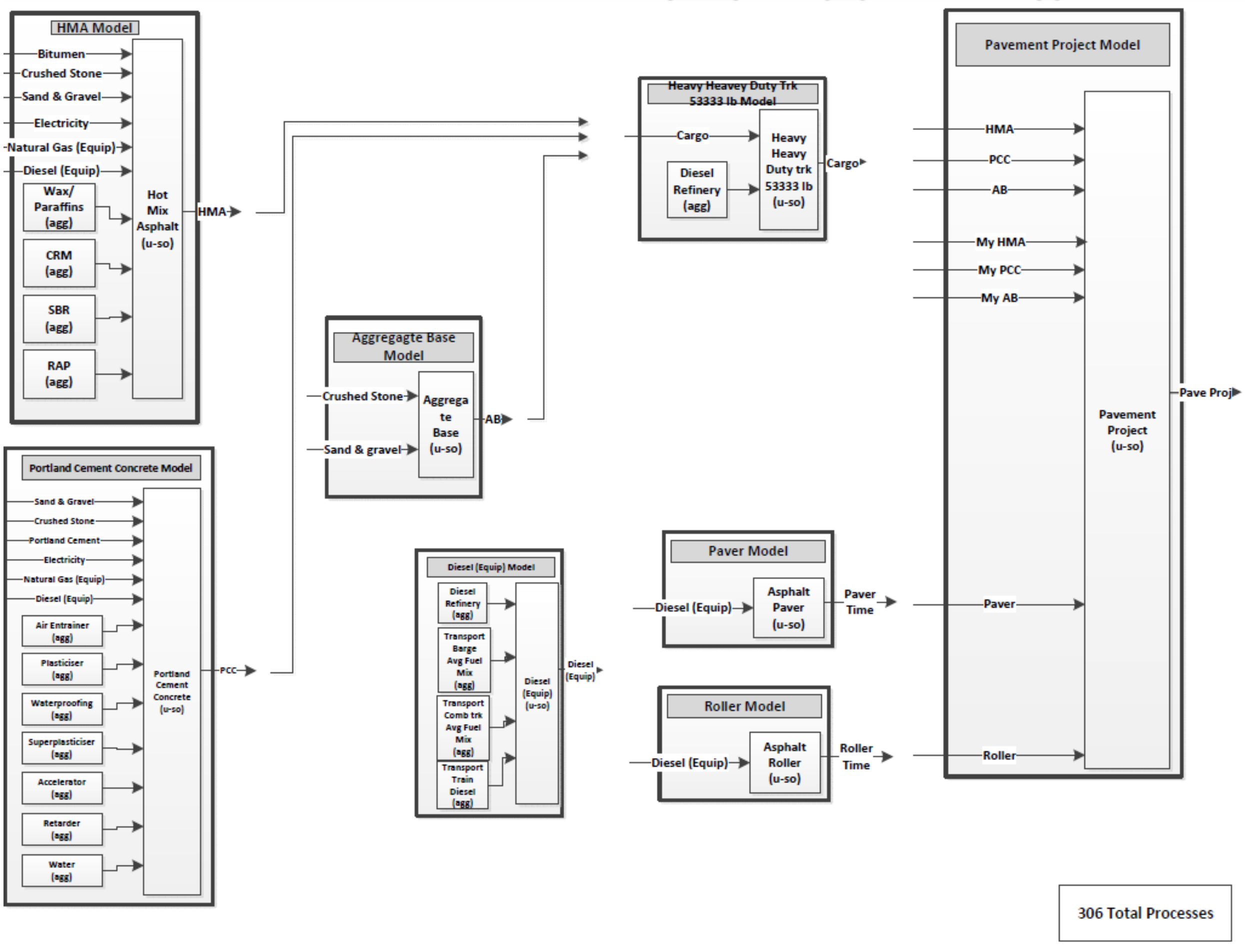


Figure A.1: Pavement Project Process Model.

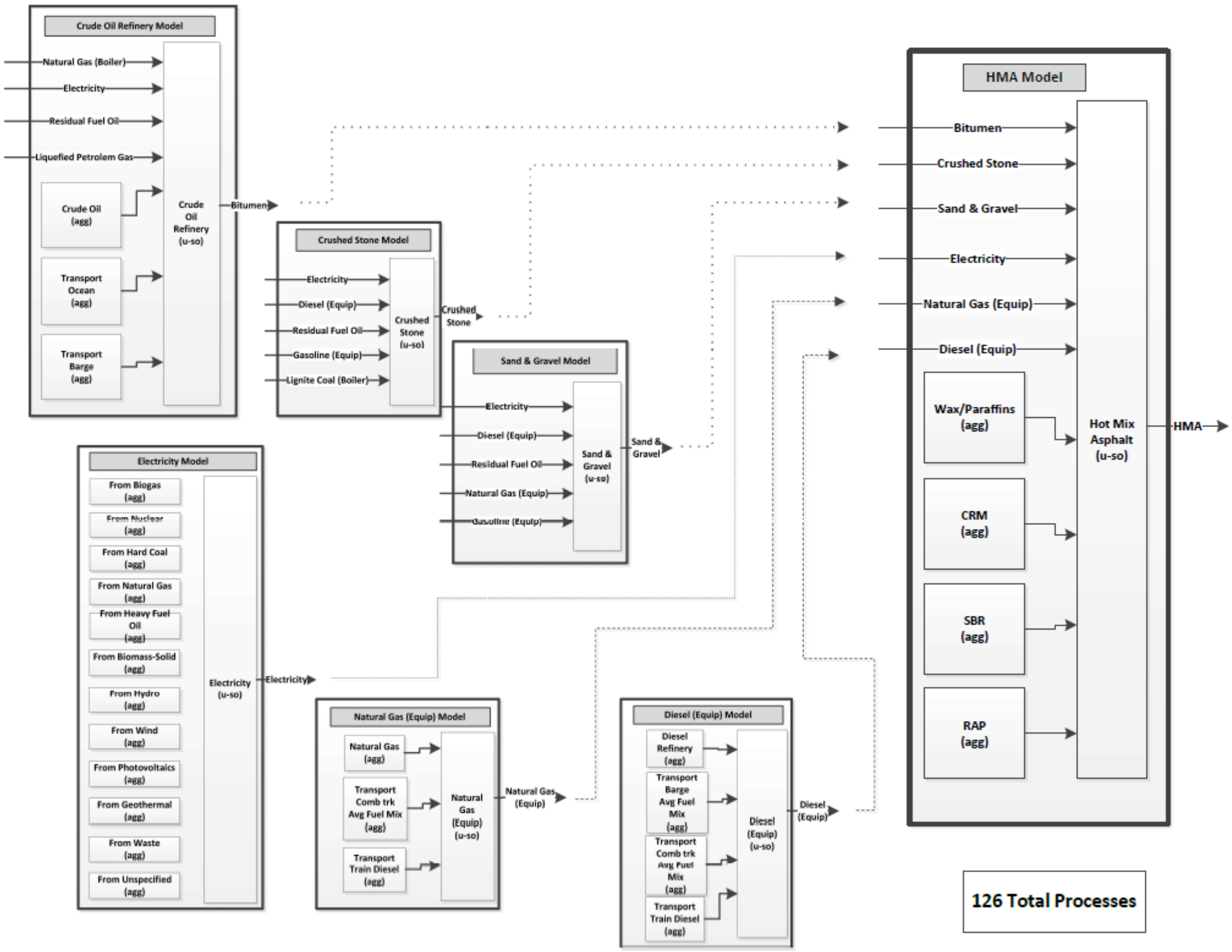


Figure A.2: HMA Process Model.

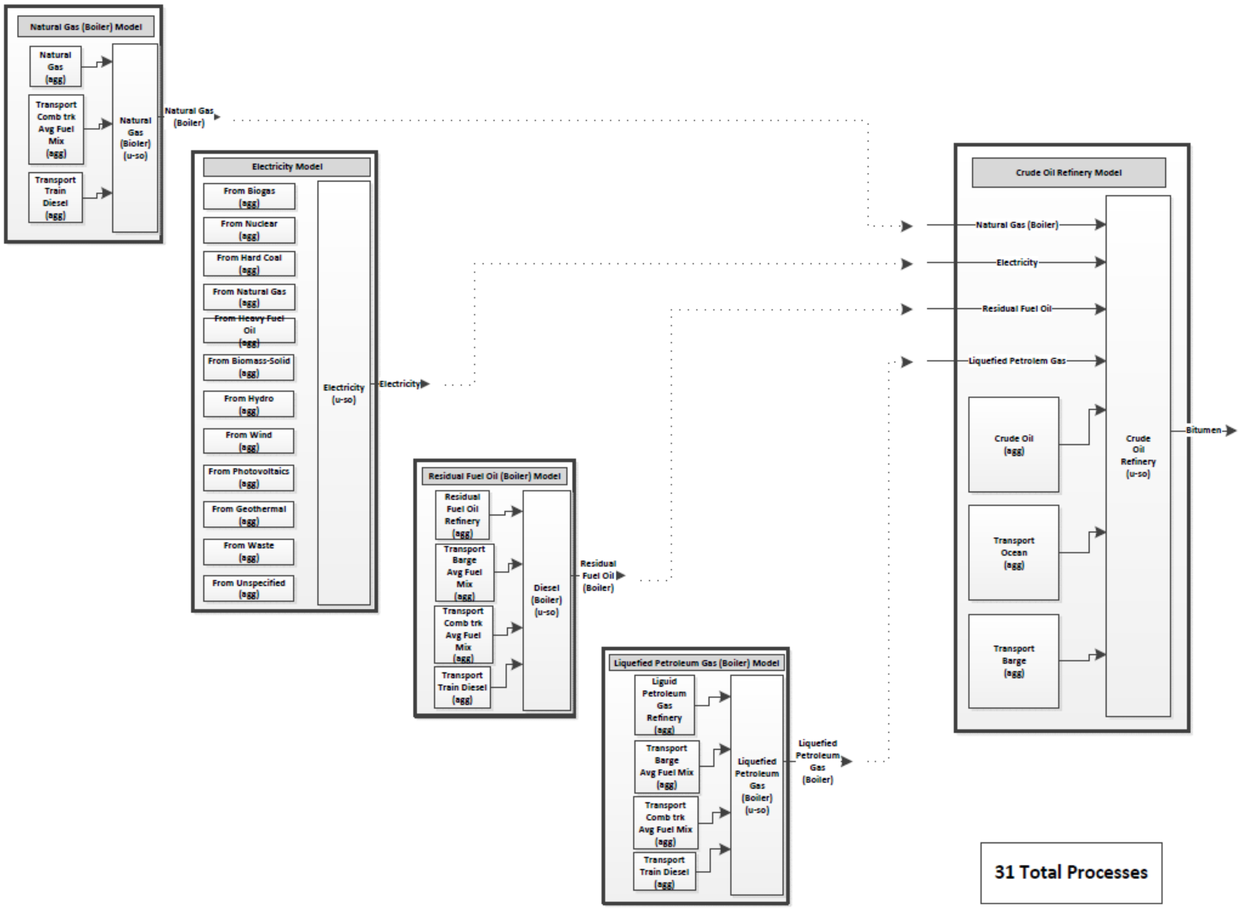


Figure A.3: Bitumen (Crude Oil) Process Model.

Stage 4

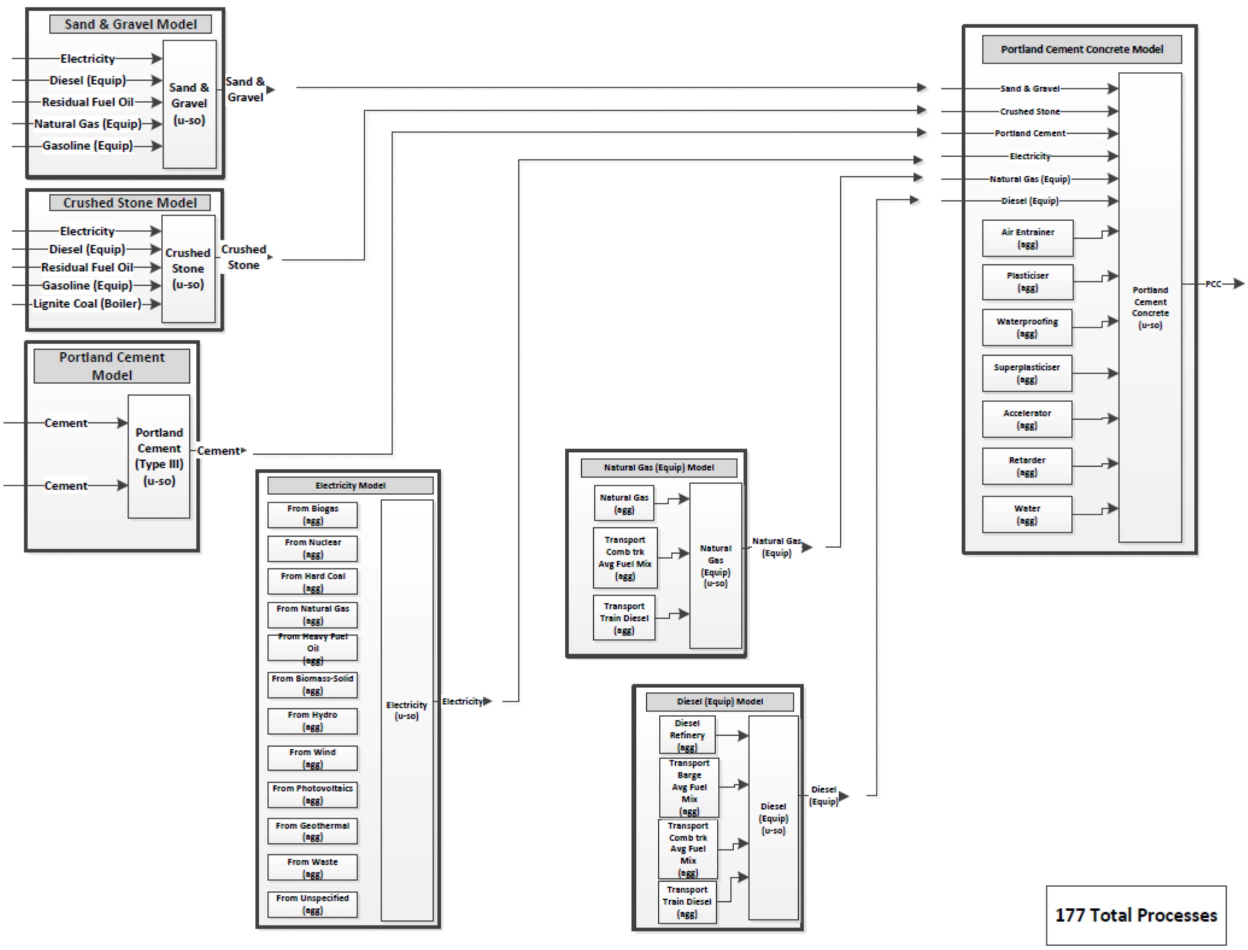


Figure A.4: Portland Cement Concrete (PCC) Process Model.

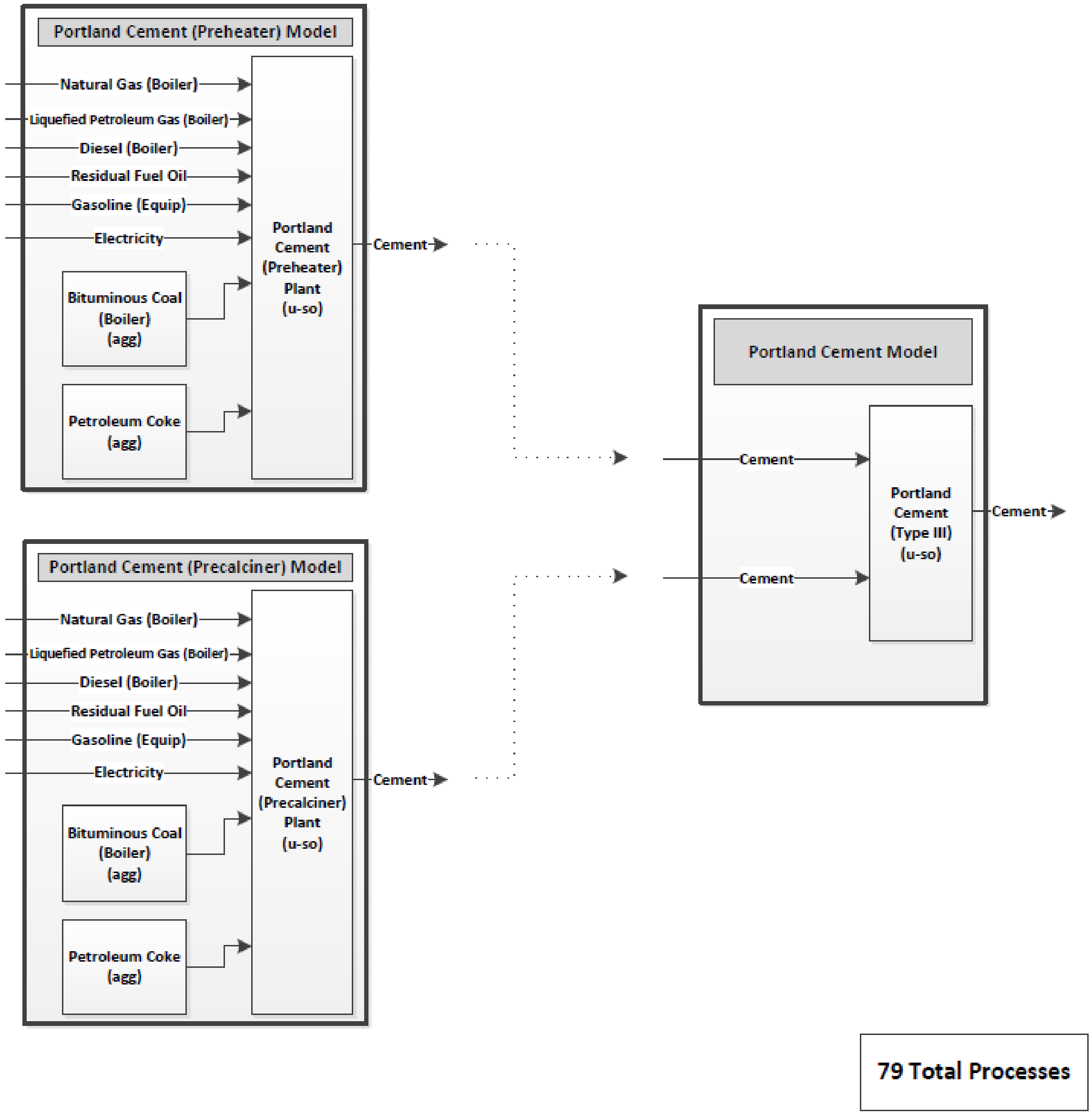


Figure A.5: Hydraulic Cement Process Model.

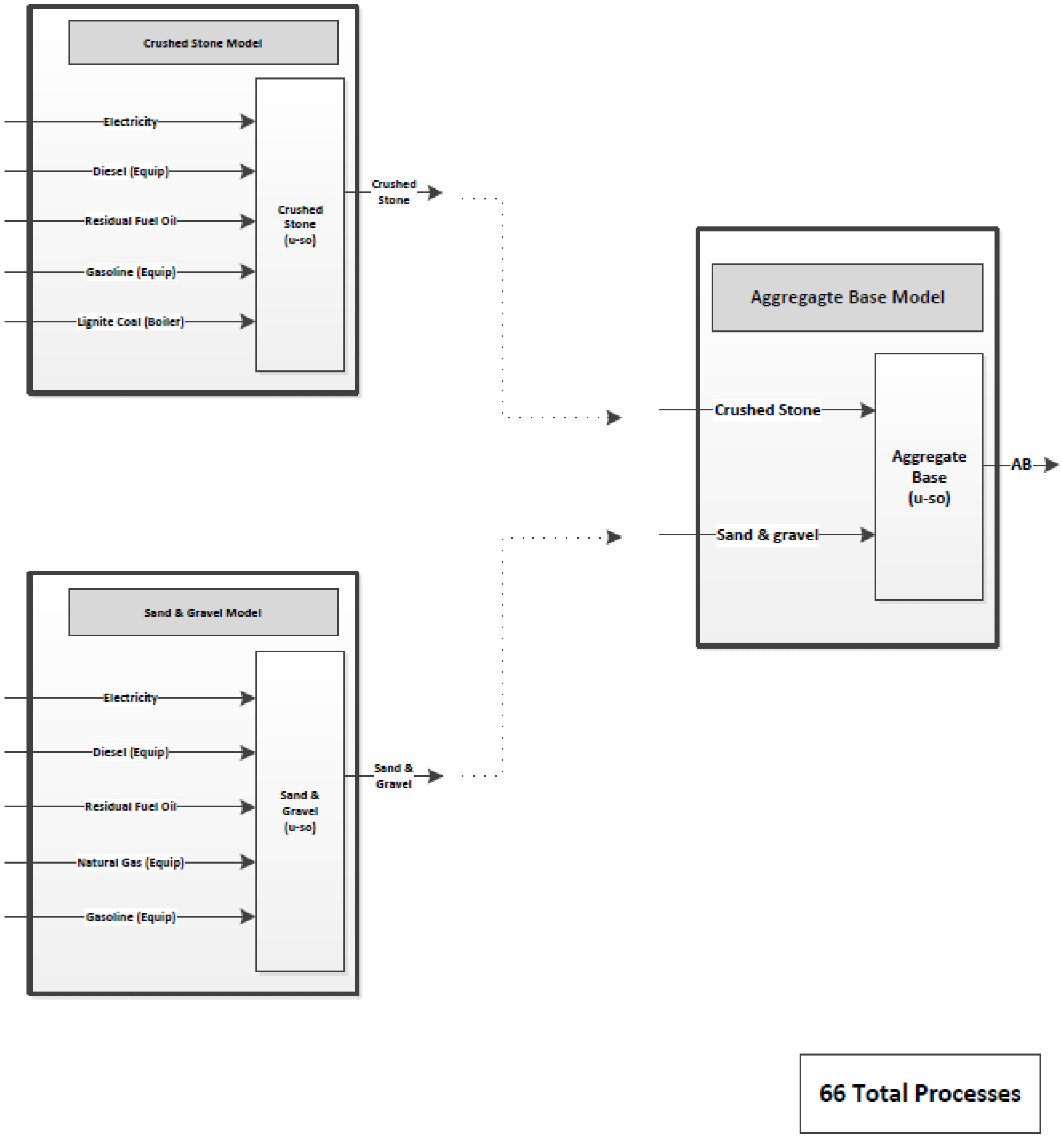


Figure A.6: Aggregate Base (AB) Process Model.

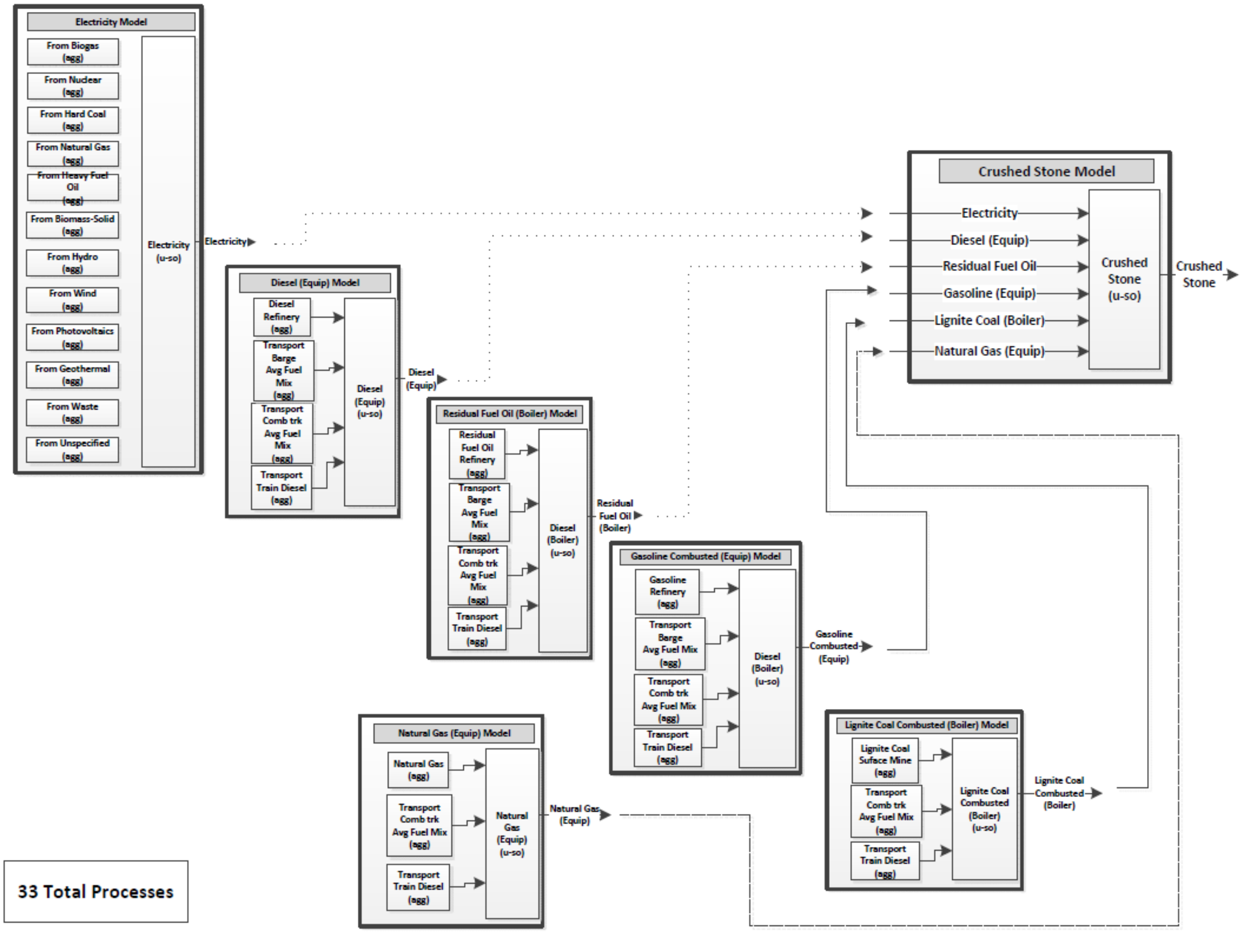


Figure A.7: Crushed Stone Process Model.

Stage 4

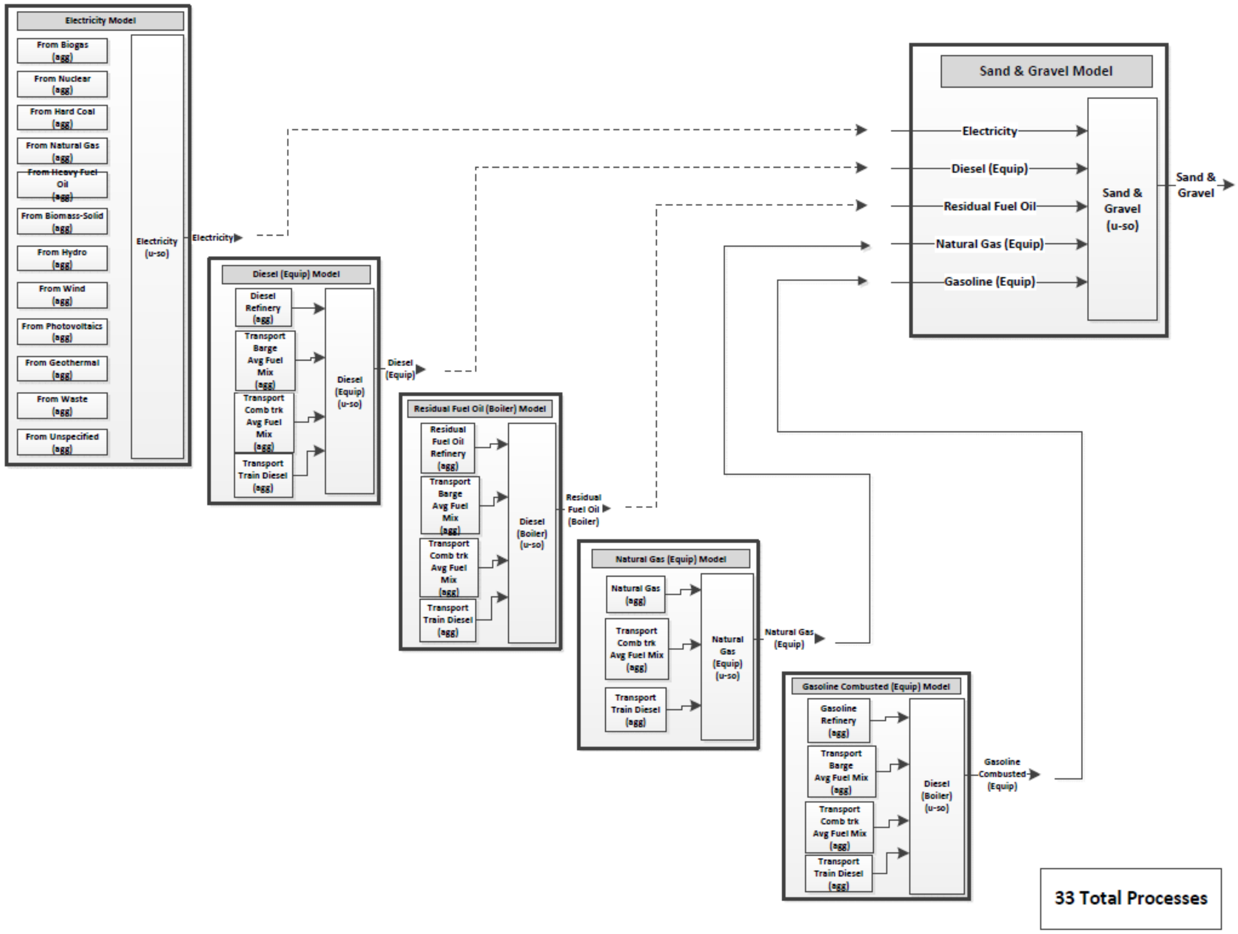


Figure A.8: Sand & Gravel Process Model.