TRANSIMS Training Course at TRACC

Transportation Research and Analysis Computing Center

Part 15

Introduction to the TRANSIMS SVN Source Code Archive and Code Structure

Dr.-Ing. Hubert Ley
Transportation Research and Analysis Computing Center

Last Updated: December 9, 2009

Unit 15

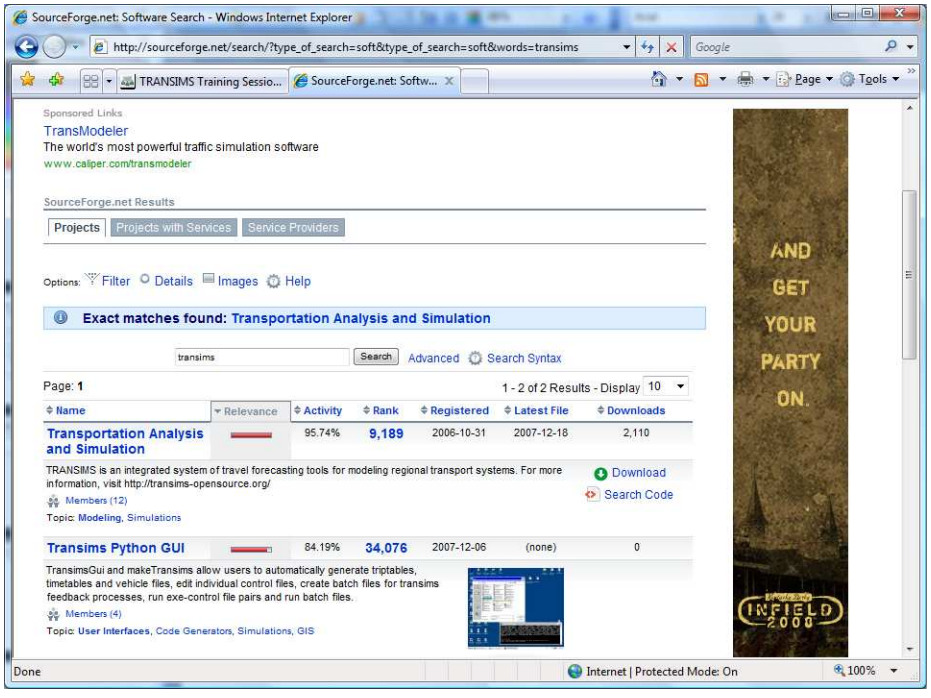
Contents

Why?

- The main reason for the existence of this lecture is to make the point that it is not so hard to look at he source code of TRANSIMS after all
- TRANSIMS is an Open Source Project, so participation of the community is essential
- The code is still heavily under development, and the documentation happens to be sparse and sometimes outdated
  - Yes, even this training course is by now probably somewhat outdated
- The training course also intends to cover in the future esoteric issues like coding conventions, collaboration issues, etc.

How to Find TRANSIMS

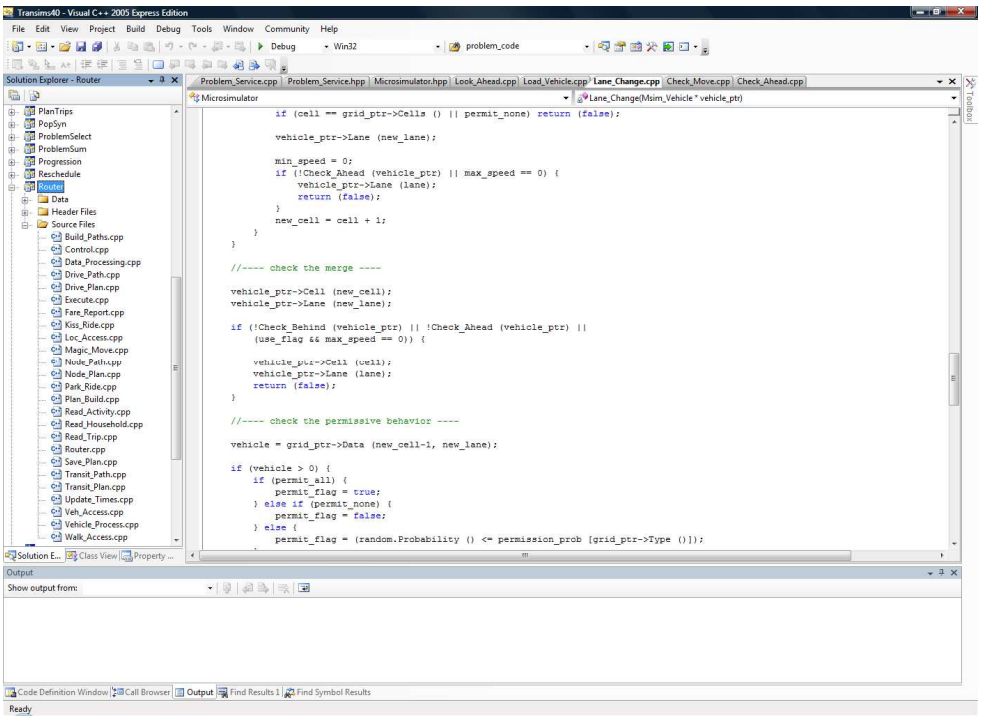
- TRANSIMS can be found at SourceForge
- TortoiseSVN is used to check out a copy
- The Visual C++ 2005 Express compiler is free from Microsoft
- Linux uses GCC



TRANSIMS Structure

- TRANSIMS is well-structured on each platform
  - Windows / Visual C++
    - A single solution file contains the entire software suite
    - Each project is included in this solution, including all dependencies
    - All compilation properties are pre-defined
    - The only thing the user has to do is a “Build”
    - That even allows to build readily 64bit versions
  - TortoiseSVN is a visual agent to simplify keeping your working copy updated from SourceForge
    - Regular updates ensure that you have the latest and best version if you desire to take the risk
    - The TRANSIMS project team has a free license for the VisualSVN software that allows integrated source code archive access
  - The user community has developed a lot of expertise and helps each other through use of the TRANSIMS forum

Visual C++ 2005 Express and TRANSIMS



TRANSIMS Layout

- There is one library containing all the functions that are needed by more than one tool (SysLib)
  - Writing and reading files
    - Many different formats, including some databases, GIS layers
  - Error handling
  - Progress reporting
  - Problem reporting
  - Network loading and processing
  - Demand file loading and processing
  - Control file processing
  - Coordinate system reprojections
  - Record management and indexing
  - String processing
  - and much more …

Useful Source Code Sections

- For novices, the number of source code files may look threatening
- But there are commonalities between all projects
  - Control.cpp
    - This is the first thing to search for in each project. The processing of the control keys for the executable is programmed in this file, and the user can look up a few dependencies and values if in doubt
  - Execute.cpp
    - This is a common file in each project that serves as the main starting point for all the actions of the particular program
  - The .cpp file with the name of the project, e.g. Router.cpp
    - Settings of many default values for variables representing control keys in the code
    - Lists of control key strings
    - Also look at the corresponding header file, e.g. Router.hpp

Useful Source Code Sections

- To simplify finding important header files, all system-wide header files are located in the "**SysLib/include**" directory
- To find information about file formats and field names, take a look at the
  - "**SysLib/Network/Files/Source Files**" directory in the solution explorer as well as the "**SysLib/Demand/Files/Source Files**"
  - These files specifiy field names for Version 3 and Version 4 formats, as well as which fields are required and which ones are optional
- Searching the source code for specific terms is often useful, for example for the text of error messages. Searches in Visual C++ can be on the scope of all source files, the current project, or even just the current file
- For really tough cases, Visual C++ provides a powerful debugger that allows stepping through the source code line by line to analyze code behavior

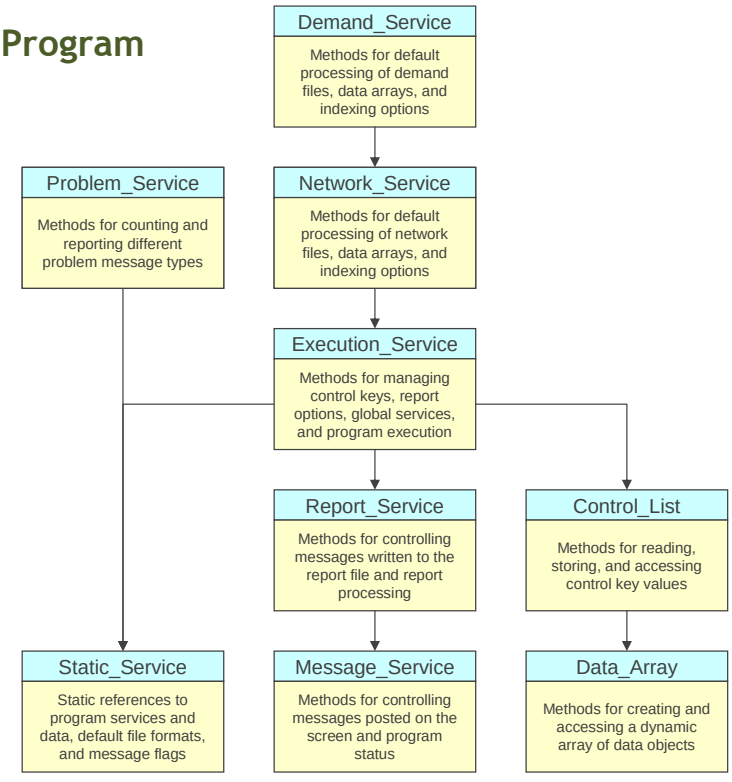
Linux Auto-Tools and TRANSIMS

- Under Linux, TRANSIMS compiles out of exactly the same directories and files as under Windows
- The typical "configure", "make", and "make dist" is being used to build "tarballs" for further distribution
- Linux executables are more specific to distributions, hardware, and configuration options for specific platforms, therefore Linux TRANSIMS is only distributed in source code format
- RPM redistributable binary packages can be build readily on appropriate platforms
- This work is mostly performed at Argonne / TRACC, but checked in rapidly for each code update released by AECOM or other contributors

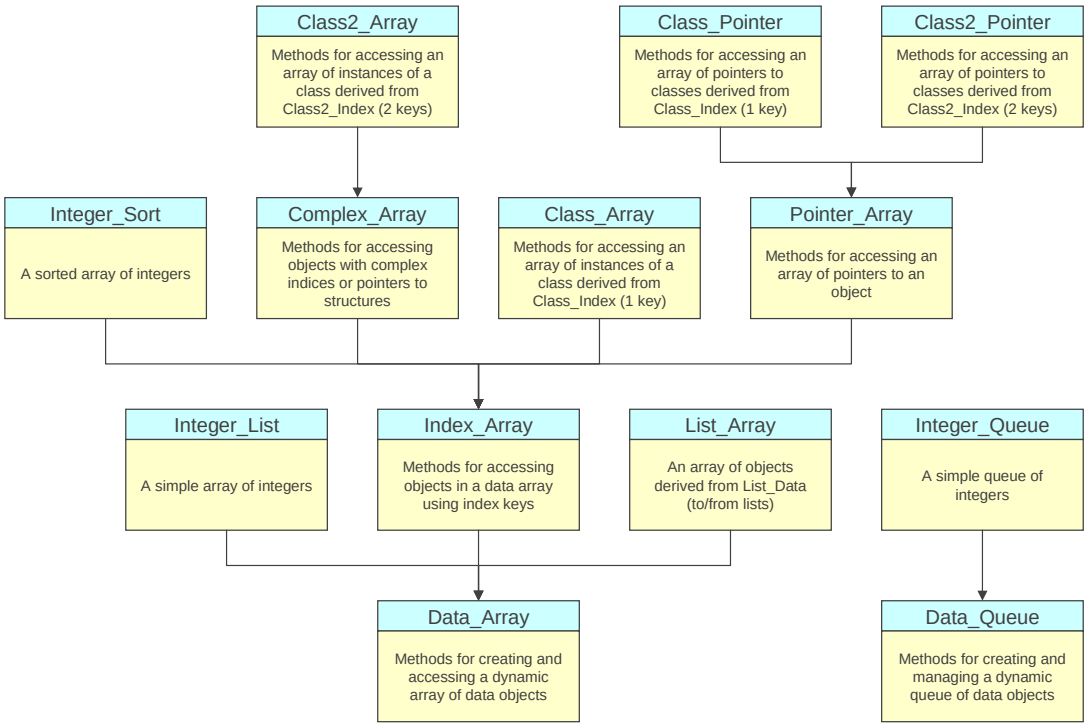
Sample Projects

- There are several “templates”, which are rather empty projects based on the TRANSIMS framework. They are checked in in the trunk of the SVN archive
  - Network Template
  - Report Template
  - Program Template

Standard TRANSIMS Program Execution Services



Dynamic Arrays of Data Objects Stored in Memory



Credits and Acknowledgements

- GIS visualization materials were mostly developed at Argonne based on the TRANSIMS tools developed by AECOM for USDOT
- Chicago road and transit network data used in some of the examples was provided by the Chicago Metropolitan Agency for Planning
- USDOT provided the funding for the development of these training materials
- USDOT provided the funding for the TRACC computing center and the resources necessary to perform these training session