

SIDRA API WORKSHOP

Melbourne, November 2018

TRANSCRIPT of Workshop Presentations and Discussions

Edited by Rahmi Akçelik

This is an edited version of the audio recording of workshop presentations and the discussion session shown in the Contents list. We recommend that you open the corresponding PowerPoint presentation when reading a section.

Updated: 2 May 2019

CONTENTS

PRESENTERS	3
1 Welcome and Introduction to SIDRA API WORKSHOP	4
Rahmi Akçelik, SIDRA SOLUTIONS	
2 An introduction to using API for SIDRA INTERSECTION 8 and VOLUMES Utility	6
Mark Besley, SIDRA SOLUTIONS	
3 SIDRA BATCH Python sample program using SIDRA API	11
Harry Cai, SIDRA SOLUTIONS	
4 Aimsun – SIDRA Interface	14
Mohammad Saifuzzaman, Aimsun - A Siemens Company	
5 Tracks – SIDRA Link	19
Grant Smith, Transportation and Traffic Systems Ltd	
6 Using SIDRA API for EMME and DYNAMIQ	22
Paul van den Bos, Freelance Transport Modeller	
7 GTASIDRA - A Process Integrator	28
Luca Rosati and Rebecca Strachan, GTA Consultants	
8 Discussions on user needs and API development	31

PRESENTERS

RAHMI AKÇELİK

Dr Rahmi Akçelik is a leading scientist and software developer with 40 years of practical, research and training experience in the area of road traffic operations, traffic engineering, management and control. He is the author of the SIDRA INTERSECTION and SIDRA TRIP software packages.

MARK BESLEY

Mark Besley is a scientist and software specialist who has worked in the traffic management area since 1980s. He has been making a significant contribution to the development of the SIDRA INTERSECTION software in collaboration with Dr Rahmi Akçelik.

HARRY CAI

Harry Cai is a Software Engineer at SIDRA SOLUTIONS. He completed Master of Applied IT and Master of IT (Minor Thesis) degrees at Monash University. He contributes to the development and support of SIDRA SOLUTIONS software products.

MOHAMMAD SAIFUZZAMAN

Mohammad Saifuzzaman (Saif) is a Product Specialist at Aimsun Pty Ltd. He completed his PhD in Traffic Engineering from Queensland University of Technology (QUT). He is a transport engineer with over seven years of international experience. He will present some use cases and the workflow to link SIDRA with Aimsun Next, demonstrating how this interface works with a hands-on example.

GRANT SMITH

Grant Smith is the Director of Transportation and Traffic Systems Ltd. He is the developer and owner of TRACKS, a land use and transport modelling system software package. He will present the TRACKS-LITE package that links the SIDRA INTERSECTION traffic model with a TRACKS transport model.

PAUL VAN DEN BOS

Paul van den Bos is a Freelance Transport Modeller with 30 years of traffic and transport modelling. He will present his experience with linking SIDRA INTERSECTION with EMME and DYNAMIQ transport planning and simulation packages.

LUCA ROSATI

Luca Rosati is a transport planner, modeller and data analyser with over 20 years of experience. He is a multidisciplinary modeller, skilled in both macro and micro approaches with experience on more than 80 projects. Previously with the University of Rome 'Tor Vergata', he has published more than 30 international papers. He will present the GTASIDRA process integrator.

1 Welcome and Introduction to SIDRA API WORKSHOP

Rahmi Akçelik, SIDRA SOLUTIONS

Rahmi: Good afternoon everyone. I'd like to welcome you all to this first SIDRA API workshop in Australia. Thank you very much for your interest. I hope you get good value out of this meeting.

Today, we have five delegates from local Vic Roads, two from RMS New South Wales, one from Queensland Transport and Main Roads, 20 delegates from 16 different consultancies, and twelve SIDRA SOLUTIONS technical and other office staff attending the workshop.

Rahmi: I'd like to ask a quick show of hands, so we can assess the type of work you do quickly. You can show your hands more than once if you wish:

- traffic modeller (including SIDRA INTERSECTION users for traffic operations, design, and signal control and similar areas),
- transport planner (including strategic transport model users),
- IT support person, software engineer, and similar.

Rahmi: We are going to do is take the questions straight after the presentation. For each presenter, there'll be 15-minute presentation time and 5-minute question time.

Rahmi: I'd like to say a few things about SIDRA SOLUTIONS.

- **Subscribe to SIDRA NEWS:** If you are not subscribed already, it's very simple to subscribe on the home page. We do not send the SIDRA NEWS email messages frequently, but they will contain important information for you.
- **Support system:** We have a ticket-based support system. API information is under the support section of the SIDRA SOLUTIONS website.
- **Tutorial videos:** You're probably good SIDRA users, but the tutorial videos are useful for people joining your organisation and asking you about SIDRA as good users. You can refer them to these tutorial videos first.
- **Training and events:** All SIDRA SOLUTIONS training workshops and other events are listed on our website. Visit the Training and Events pages. When the training programs and events are available, they are announced via SIDRA NEWS and direct mail.

Rahmi: **Documentation:** We have many papers and reports available for download on the ARTICLES page of our website. SIDRA is one of the most documented software. These include the latest documents as well as scanned copies of the original 1980's SIDRA documents.

Rahmi: We also made SIDRA documentation available on the RESEARCH GATE website about five years ago. This is a great website for exchange of research papers. The website stats (in

November) indicated more than 25,000 reads and about 1,900 citations of our papers. This is a good indicator of the strong research basis of the SIDRA software (research for 20 years at Australian Road Research Board and 20 years at Akcelik & Associates).

Rahmi: SIDRA MODEL FUNDAMENTALS workshop: This is a workshop I prepared and our colleague David Nash runs successfully. This workshop presents basic traffic modelling information for people new in our area. It is useful in using all software packages. We also recommend it to experienced modellers for a review of traffic modelling and traffic flow theory knowledge. Computers are not used in this workshop. Instead, manual exercises are included.

Rahmi: SIDRA was first released in 1984 as a mainframe computer program. Over the last 35 years, it has been developed continuously in parallel with the changes in computer technology and in the modelling needs of traffic engineers and planners. Instead of the private car focus in early days, we now have a transport focus including emphasis for bicycles, public transport, heavy vehicles, and so on.

Biggest change in the 35-year history of the software was the introduction of the lane-based network modelling as an extension of intersection modelling.

Rahmi: SIDRA is used widely in many countries around the world. The largest user groups are in USA and Canada, Australia and New Zealand, South Africa, Middle East, Malaysia and Singapore as well as Slovenia, Norway, Spain and Italy in Europe. A few months ago we were in Norway running a training workshop and a SIDRA User Group meeting. We learn a lot from different traffic engineering practises different driver characteristics around the world which enhances SIDRA.

Rahmi: The last slide is about SIDRA INTERSECTION API. This workshop will cover most usage of the SIDRA API. Unfortunately, we will not cover the TORUS program, the first software package which linked to SIDRA via its API. TORUS is a special CAD-like roundabout design package.

This is the end of my introductory presentation.

Rahmi: Next presentation is by Mark Besley. Mark will do a general introduction to the SIDRA API, then he'll go onto the second item for him, which is the volumes utility that we have prepared. And, he'll do the presentation on that, he'll continue. He says he'll do it in 15 minutes. We'll see. Then, I think we'll also get Harry to do the presentation. He wrote a special program for this, just a demo. We called it SIDRA Batch, a bit of scenario analysis. So, Harry will present that. Then, we'll take questions on those, then we'll continue.

Rahmi: Mark, if I may invite you.



2 An introduction to using API for SIDRA INTERSECTION 8 and VOLUMES Utility

Mark Besley, SIDRA SOLUTIONS

Mark: I'm going to give a broad overview of the Application Programming Interface (API) for SIDRA INTERSECTION. We'll go into more details later on. In some cases, you'll know some of this stuff. But, to get everybody on the same page, I'll do this general description.

Mark: I'll talk about:

- different ways we can use the API,
- what sort of resources are available to you mainly via our support site,
- what we're heading towards in the future, though of course that might change if we get lots of good ideas at this workshop,
- and, just point you in the direction of examples and sample code.

Mark: So, what's the Application Programming Interface? It's a way of providing a programming interface at the source code level, so that you can have an application you've written that will communicate with SIDRA. And, as we're saying here, that happens outside of the SIDRA user interface. You don't need to run SIDRA on the screen necessarily to be using the API. This can all be happening quite invisibly if you like, in the background. And we'll see examples of that.

Mark: Now, I'll talk about another thing that's not really API, in that it doesn't fit the strict definition of an application programming interface. You can access the SIDRA input and output database in the SIDRA project file (.sip8 file) that you're using. Possibly some of you realise it's an SQL database. We're not locking it away from you, or trying to hide things. It's not password protected or anything like that. You can access it directly. So, I'll talk about that as well.

Mark: So, what types of things does the Application Programming Interface allow you to do? Some of these words are out of our user guide and website and so on. But, just a quick summary of examples. You can programmatically get access to all of the input and output data. It's a pretty big range of data, and you can do what you like with it. You might be wanting to get your own input data directly into SIDRA somehow without having to use the interface, or you might be wanting to pull out your own outputs for reports and so on.

Mark: Beyond that people had broader needs. So, we've added things like the ability to create project files, create sites and networks and routes within those projects. Specify geometry, add a leg, remove a leg and so on. And, specify any other data you want. Phasing and timing data, geometry, all those things. Having done that, of course the ability to process is fundamental. You've created a whole lot of inputs, you need to be able to tell the API to process that and generate a set of outputs.

Mark: The three options to use the SIDRA API I'll talk about are:

- Component Object Model (COM),
- programming interface via Microsoft .NET, and
- direct access to project file database.

Mark: You've probably heard of Component Object Model. I'll just refer to it as COM. I'll talk in a bit of detail about the programming interface via COM and Microsoft .NET to make it clear what the distinction is between these two interfaces. I'll also be talking about direct access into the project file database.

Mark: COM was introduced by Microsoft a long time ago in 1993. It's a binary interface standard for software components. It provides a stable application binary interface (referred to as ABI). Now, this is talking about binary data rather than an application programming interface. COM is about communicating at the programming source code type level, ABI is about the binary interface. But, let's not worry too much about that. I think the point of using COM is that it takes care of all that stuff for you, and we use the COM interface. We'll let the binary stuff take care of itself. So, let's not worry too much down at that level.

Mark: Probably the main thing to think about in terms of the interface options available to you is, which one is suitable for the programming that you're doing. And that really depends on what sort of programming language you're using. Some examples of programming languages for using COM interface are C, C++, Visual Basic. Also, Python has become very popular of late, and a lot of people are using that. Harry is going to demonstrate a little Python program after my presentation. I'll look at Visual Basic in the context of our Excel VOLUMES utility.

Mark: Now, another fundamental thing about either COM, or the .NET interface I'll talk about in a few minutes, is the fact that you need a licenced copy of the SIDRA INTERSECTION software on the same computer as you're using. When you run the API program, the first thing the API is going to do is to look if SIDRA is installed. If it is, it will check if the software is licenced? If it is, it will work. If it doesn't find a licenced copy of SIDRA, because either the software is not installed at all, or there is not a valid licence found, the interface won't work. You'll get some error messages.

Mark: The other way that we can access the API language-wise, strictly talking about a true application programming interface, is via Microsoft .NET. You've probably all heard of .NET. Maybe you're familiar with it, maybe not so familiar. It really depends on the style and type of programming that you do. Microsoft introduced .NET more recently in 2002. Their original idea was language interoperability.

Mark: You might be aware, it compiles into a common language runtime. You could write in three or four different languages, and sort of have them all glued together. And, it also simplified a number of programming tasks. They install the .NET framework on your Windows computer, and that takes care of things if you use .NET. It takes care of things like memory management, making sure versions are correct, you're calling up the correct version of libraries, and things like that. There are a lot of .NET languages available, but C# is by far the most commonly used one.

Mark: All of this fancy version control has a lot of benefits, but it did create a bit of a headache for us. Because, we found that every time we released a new SIDRA update, the internal

versioning would change, and potentially the API would stop working. We came up with what we call a "late binding DLL" solution which we provide for you. It can be downloaded from our website. You will need to use that to keep your .NET API application running. Again, the same restriction applies. You need a licenced copy of the SIDRA software on the computer that you're using.

Mark: I'll now move to the SIDRA database. This is not strictly application programming interface, but it's a way, I think that a lot of people would be interested in accessing the SIDRA input and output data. Currently, it's a compact SQL database. We are moving to a different SQL, SQLite in SIDRA INTERSECTION Version 9. In this method, you're working directly with the SIDRA project file. You don't have to have SIDRA on your computer. It's very flexible and powerful in that way as you can just get in there and change anything you like., but of course accessing the SQL data directly in this way means there's no error checking.

Mark: Obviously reading data out of the database is not a problem. But serious errors may result if you start getting in and writing new data and thinking "I'm going to be really clever. I'm going to push that data straight into the database." I'd really recommend, keep this very limited to things like volume data, or something that doesn't interact with other data, like changing lane widths or some sort of basic data like that.

Mark: If you start trying to do things like removing a leg of an intersection, by operating on the SQL, you're not going to know all of the database relationships that you should take care of. You'll almost certainly run into trouble. Probably end up with a project file that is no longer usable. On the other hand, when you access via the API to say, delete a leg, the API functions take care of all the housekeeping. They get rid of all the data that's no longer relevant. So, just be very wary about the sort of things you might think of doing directly with SQL.

Mark: I'll talk about some limitations in the API and some precautions that you should take. Firstly, pictures we get asked about a lot. You can get the site or network layouts directly through the API. We don't currently provide support for other graphical outputs, things like the movement displays for example. But, that might be something we can discuss later on.

Mark: Don't access the same project file simultaneously through multiple programs. You'll almost certainly get into database conflicts, potential data corruption and so on.

Mark: Currently, we don't have restrictions about site names, site ID's. Again, this has been discussed a bit. We allow anything, including duplicates. If duplicates is a problem with your API application, you need to take care of that, and ensure it can't happen. We're not providing restriction on that at the moment. We might, again, talk more about this.

Mark: Some people have also wanted to run the SIDRA intersection user interface via command line. You can run it via a command line. You can open a project that way. There's no method currently to specify which site you want selected after opening it in that manner.

Mark: So, where is your first point of call if you want to get into all this stuff? I would recommend, get onto the SIDRA support website, <https://support.sidrasolutions.com>, or use the support link from our main website. "SIDRA UTILITIES and API" will guide you into the right part of the support website. You'll find a very comprehensive CHM type help

file. Download and start it up. It gives you a complete listing of Classes, Properties and Methods. There's hundreds of pages like this of stuff. So, huge wealth of information in there. This will also help you understand the database structure of the project file (SIP8). We're using the same conventions in the COM interface and in the database. So, if you want information about either, this document should be helpful.

Mark: Beyond that, if you get really stuck, you can approach us directly for help. Of course, remember the resources we have to provide assistance are fairly limited. We're certainly more than happy to help you with specific functions, classes, methods and so on in the API, but we cannot write the application for you.

Mark: Heading into the future, we know there is a lot of enthusiasm from users. Just the fact that you're all here today tells us about how enthusiastic users are. We will continue to develop the SIDRA API and provide support in the future. At the same time, we will try to maintain a similar structure in the API and minimise changes that might break your program. If you've got a SIDRA INTERSECTION Version 8 API program working now, you'll probably have to change a couple of the calls to Version 9 when it comes out. We would hope most of the rest of it would continue to work.

Mark: As I mentioned before, the SQL compact database we're currently using will change to SQLite. It should obviously only affect programs that access the project file directly. But, depending on how your data layers and your own program handles this, it might be a relatively small change. Just be aware that this change is definitely happening.

Mark: We're going to continue on now with a couple of demos. On the website, there are example programs in C#, C++, Python and Excel using VBA. I'm going to give a demo of the VOLUMES Excel utility right now. And then, Harry's going to continue on afterwards, and talk a bit about the API, and show you a demonstration in Python.

Mark: I know some of you actually got into VOLUMES and have changed it and added more functionality. But, who's at least used or looked at the VOLUMES Excel? [About 10 people showed hands.].

Mark: I'll give you a quick demonstration now of what the VOLUMES utility is about. On my desktop, I've got an SIP8 project file called "Test VOLUMES" and I've got a link to the VOLUMES Excel utility. So, I'm going to start it up.

Mark: Now, this is running, but if you run it, it'll probably have an error message along here saying "This has got macros in it, and you need to enable them to run." Before I started this, I went into the Trust Centre and turned on the various options to allow macros to run without any warning messages. But, just be aware, there may be protections built into your system against running Excel macros. Depending on how your IT is managed, you may need to get your IT administrators to allow this. So, in your default system, the VOLUMES utility might be blocked out but that's a simple enough change.

Mark: Here are the Introduction and Licencing sheets. But the main place we want to go is the Control sheet. I'm going to open a SIDRA INTERSECTION project file using a couple of buttons here. I'm going to find my Test Vol SIP8 on the desktop. This loads up. It shows me the name of the project file and the sites in it in a dropdown list. We've only got one site in this project for simplicity. If there were 10 or 100 or whatever, there'd be a big list there. You would select one site.

- Mark: As a demonstration, I'm going to use the Get Volumes function. Just before I do that, I'll go to the Volume Input sheet. Here it's showing light vehicles with volumes of 1 . That's just the default setup in a new copy of the VOLUMES Excel. This is showing the volume from South to East, South to North, South to West and so on. It is all setup in this OD format, allowing for the eight possible legs that we allow in SIDRA.
- Mark: I'm going back to the Control sheet and click "Get Volumes". It sits there for a few seconds processing. Then opens the Volume Input sheet and shows the volume data that it retrieved from that project file. So, you can see all volumes here. South to East Right turn 70, South Left 115, South Through 500 and so on. So, that's data straight out of the file.
- Mark: But, the beauty of this Excel is that it allows me to change this data. Now, just as a demo, I can go and type some numbers. For example, 123, 456, 78 (probably a bit more reasonable for a turn movement). We could type them all in. But, rather than typing, a simple Excel macro or even some formulas could easily put data into this sheet from another source. That might just be another Excel page. You could have all your 15-minute counts in some format in Excel and just link them up to here. There may be a simple macro "bring in the PM volume", something like that. Populate this, get a set of volumes, or use one of the Excel data tools to access a database or other sources like that.
- Mark: Now, having put those in, I can go back to Control, click "Update Volumes and Process". I haven't got the SIDRA interface open. Excel has gone off, taken those volumes, put them into the SIDRA project file, and invoked a SIDRA run. After processing is finished, it's come back with a set of outputs. We've got some outputs here in this VOLUMES Excel. The Intersection Output sheet is like the Intersection Summary, Lane Output like the Lane Summary, and so on. These are the same information I would see in the SIDRA user interface. But, we never saw SIDRA running. It all happened in the background via the API.
- Mark: Now, I'll go back to the Control sheet, and I'll click "Save Intersection Project File". I'll then close the Excel file. This prompt is asking about saving the Excel file itself. I don't need to save it.
- Mark: I will now run SIDRA INTERSECTION 8. I will load the project and, if I look at the Volumes Input for South approach, I see 123, 456, 78. Those are the numbers I typed into the VOLUMES Excel, because I saved the project file before I closed Excel, they're now permanently part of this SIDRA project file. If I run SIDRA, I will see the same output numbers in Lane Summary, Movement Summary and so on, as I saw in Excel.
- Mark: Now, I will close the VOLUMES utility and invite Harry for his presentation.



3 SIDRA BATCH Python sample program using SIDRA API Harry Cai, SIDRA SOLUTIONS

Harry: I'm going to introduce a small Python program called SIDRA BATCH to demonstrate how to use the SIDRA INTERSECTION API through the Microsoft COM interface. I will first discuss the Project files I prepared for this example, and then discuss the Python source code.

Harry: In this example, I prepared four Project files, one is for each Proposed Design for an intersection. In each Project file, there are two Sites., one with AM Peak volumes, and another with PM Peak volumes. For each Site, I specify a Category, which is a string value, describing the Proposed Design. I will use this parameter to categorise Site outputs into different groups.

Harry: Let's have a look at the Project file named "Signals - Scenario A". In this Project, I set the Category as "Signals - Scenario A" for both AM Peak and PM Peak Sites. Scenario A doesn't have slip lanes, and Scenario B has slip lanes. The Python program will process the Sites in all Project files, read and save all output data as the local Python data, and generate a Project Summary Report.

Harry: Let's have a look at the **Python source code**.

[For the purpose of clarity, PowerPoint presentation slides on Python source code are copied below.]

Python Source Code

❖ Prerequisites

- ❖ Install Python for Windows Extensions Package "pypiwin32"
 - ❖ Install through PIP: run the command "pip install pypiwin32", or
 - ❖ Download the package from <https://pypi.python.org/pypi/pypiwin32>
 - ❖ Note: There is a similar package called "pywin32". Do not use this package.

❖ Prerequisites

❖ Prepare SIDRA API-Python Package "SIAPIComPy.py"

- ❖ Download Python Sample program from SIDRA Support website -> Resources home -> SIDRA INTERSECTION -> SIDRA UTILITIES and API. The package is included in this sample program.
Or
- ❖ Use the Python tool "makepy.py" to generate it.
 1. Open "Command Prompt"
 2. Run "<Python_install_folder>\Lib\site-packages\win32com\client\makepy.py"
 3. Select "SIDRA INTERSECTION 8.0 API (8.0)" in the popup "Select Library" window, click "OK"
 4. Copy the generated "A073B4B7-AFE5-4EE8-A7D3-6FFEB4FFEB09x0x8x0.py" file to the folder of your Python SIDRA API Program file and rename it to "SIAPIComPy.py".

❖ Process Sites from Multiple Project Files

❖ Import required packages

```
import win32com.client as w32c
import pythoncom
import SIAPIComPy
```

❖ Create SIAPI object, wrap it with `SIAPIComPy.ISIAPI` class

```
self.siapi = SIAPIComPy.ISIAPI(w32c.Dispatch("SIDRASolutions.SI.API.SIAPI"))
```

The wrapping is important. Without it, Python does not know what attributes can be achieved from SIAPI object. We need to use wrapping on every object that comes from SIDRA API.

❖ Open a Project file and get the Project object

```
self.siapi.OpenProject(fileName)
project = SIAPIComPy.ISIAPIProject(self.siapi.Project)
# wrap it with ISIAPIProject
```

❖ Process Sites from Multiple Project Files

❖ Get a collection of Sites from the Project

```
sites = SIAPIComPy.ISIAPISites(project.Sites) # wrap it with ISIAPISites
```

❖ Go through each Site by index

```
for siteIndex in range(sites.Count):
    site = SIAPIComPy.ISIAPISite(sites.Item_2(siteIndex))
    # site index starts from 0
```

❖ Process the Site and get the Site Output data

```
site.Process()
if site.Outputset:
    outputset = SIAPIComPy.ISIAPIOutputset(site.Outputset)
```

❖ Read Site Output Data

- ❖ Read Intersection Summary Output data for all Vehicle Movements on the Site

```
outputSiteVeh = SIAPIComPy.ISIAPIOutputSiteVehicle(outputset.OutputSiteVehicle)

myUIData.Flow_total = outputSiteVeh.Flow_total
myUIData.Capacity_effective = outputSiteVeh.Capacity_effective
myUIData.Deg_satn = outputSiteVeh.Deg_satn
# ...

self.UIProjectSummarySiteDataList.append(myUIData)
```

- ❖ The object “myUIData” is a local defined object to hold a copy of the Site’s output data, and then is stored into a local List UIProjectSummarySiteDataList.
- ❖ When SIDRA API opens another Project, the current Project Output data will be removed. We make local copies of our interested data so that we can have the data from different Projects in the list.

❖ Generate Project Summary

- ❖ Go through a list of local copied data to generate a CSV file. The final output data in this example are grouped by the Category value.

❖ Clean up

- ❖ In this example, we use Python code “with ... as ...” to create the example object pySIAPI, so that when the program finishes, its __exit__() method is called to clean up. Finally, call pythoncom.CoUninitialize() to release the COM resources.

```
with PySIAPI() as pySIAPI:
# ...
def __exit__(self, exc_type, exc_value, traceback):
    del self.siapi
# ...
finally:
    pythoncom.CoUninitialize()
```

Harry: That's the end of my presentation.

Rahmi: I'll just mention that, we have a Project Summary feature in SIDRA INTERSECTION Version 8 but it is for output within the current Project. On the other hand, the SIDRA BATCH application can create a Project Summary by using data from different Projects. So this Python program will help if you need data extracted out of several Projects because that's how you organise things, i.e. if you don't put everything into one Project.

Rahmi: The next presenter is Saif. He'll do a presentation on the link between the Aimsun microsimulation package and SIDRA.



4 Aimsun – SIDRA Interface

Mohammad Saifuzzaman, Aimsun – A Siemens Company

- Saif: Good afternoon everybody. My name is Mohammad Saifuzzaman. You can call me Saif. I work as a product specialist in Aimsun. Today, I will show you the newly built interface between Aimsun and SIDRA Version 8. First, I'll tell you something about the company because recently we have been through major changes. I will show you how the signals in Aimsun works, and then how the interface with SIDRA works, and discuss some of the future improvements we are thinking about.
- Saif: Many of you know that Aimsun was previously named as Transport Simulation Systems and our main product was Aimsun. Now our company name is Aimsun and our two products, Aimsun offline becomes Aimsun Next, which simulates all the traffic modelling offline, and the other product is Aimsun Live, which allows you to run Aimsun real-time. This rebranding happened in November 2017. We are now a standalone entity, fully owned by Siemens.
- Saif: The software comes with a list of options that you can improve, or you can use it in different aspects. Within the software, if you want to change something that doesn't come in-built, you can do scripting which is in Python. Then there is the API functionality, which I will present today. It is used to transfer information between Aimsun and other software.
- Saif: Now let's see how the signals and Aimsun works. Aimsun has a different signalling approach than SIDRA. Aimsun doesn't optimise signals. It replicates signals from different sources. Because it is a simulation program not an optimization program, we need something like SIDRA interface for our users to optimize signals.
- Saif: Aimsun can replicate almost all signal types. It can have fixed signal control plan (CP), actuated signal control plan. For the actuated CP, it follows the NEMA standards. Also, you can have actuated pedestrian phases, and you can put public transport priority. Here is one example of a complicated signal CP. This SCATS signal CP has 15 stages, 28 signal phases. It is combined of six nodes and 12 actuated signals. This type of complicated signals are possible to replicate in Aimsun. Aimsun has a rich interface with SCATS, which can be used to get the historical signal data from SCATS.
- Saif: We can import the signal geometry, signal groups, detector data, direction counts, historical signal timings. Also, we are working with RMS to update it for the SSDI format, they are making now. We also have an interface with SCATSIM. This one doesn't have to work with the historical signal timings. The SCATSIM engine will generate signal phases on the run, which is very important for our Aimsun live applications. Any node in Aimsun can be transferred into a signalised intersection. By default, we have five different node types available. You can have uncontrolled node, which is normally used for the priority intersections. If you have a fixed signals plan, you use the fixed type. So here is the type, you change in the control plan option.

- Saif: We can have uncontrolled, fixed, actuated signal control plan. If you want to do your own signal plan outside as a file, you can use that. You can use an API to load it during the simulation and if nothing is specified, it's just a normal node. Another interesting feature in Aimsun is this master control plan, which combines all the timings of the signals. For example, for each 15-minute period, this particular control plan will operate, for the next 15 minutes there will be another control plan. In master control plan, you have the option to select at which time period which control plan you want to use.
- Saif: Now I will show a few differences with SIDRA. Maybe there are more, but I am showing ones that I have found while developing this interface. Firstly, we use signal group. If we see that several traffic movements always go together, then we can group them under one signal group to simplify things. For example, when there is a through movement going in the westbound, and the left turning always comes with it, we can bring them in a signal group and in the control plan, we just have to select this signal group. We don't have to select all the turns separately.
- Saif: Another issue is the red percentage, which indicates the percentage of yellow time the vehicles will consider as red. From the simulation perspective, we need to know when we have to stop the vehicles for a particular signal phase. So, yellow time is treated as a sum of red time and green time. A red-percentage value of 50 means half of the yellow time will be used as green time and the rest will be added to the all-red period. You may adjust the red-percentage according to the observed behaviour.
- Saif: This red-percentage gave me some hard time during this interface design because SIDRA uses end gain. The red-percentage is similar to the end gain. Another thing which is handled differently in SIDRA is the inter-phase (intergreen in SIDRA). Inter-phase in Aimsun is a summation of the yellow and red time. So, every phase is followed by an inter-phase. For example, here you see that green phase followed by a dark green, which is the inter-phase. Every conflicting phase should have an inter-phase that allows the yellow and red time.
- Saif: Aimsun considers inter-phase as a separate phase. Meaning that you can have a movement in the inter-phase. For example, in this phase one, we have these two through movements and phase three we have this right turn coming. But this through movement is still going on. A four-phase signal will have actually eight phases with all the inter-phases. You can't directly transfer these phases to SIDRA. We need some kind of manipulation to bring them correctly in SIDRA.
- Saif: Another issue that is still troubling us is the multiple nodes in a signal phase. Normally when we see that two nodes come under the same control plan, for example here, we merge the two nodes and bring them under the same control plan. If you don't merge, it doesn't come under the same control plan. But in SIDRA, you do it without merging. You do it using the Common Control group (CCG). This is still in development.
- Saif: Now, let's see the video.

- Saif: I'll just briefly show you how this interface works. Firstly, you need a blank SIDRA file. You have to create a blank SIDRA file. Then go to the simulation and run your simulations to generate the volume. Once the volume is generated, you are ready to export. You go to the export > SIDRA intersection and you can select the blank file you have created earlier. Then the volume, the simulation, then you will select the node. You can select all of the nodes or any particular node. Select the vehicle type. Use car and truck. Select from which time interval the volume will be generated.
- Saif: The interface will automatically take the base master control plan from the simulation that you have run. Once it's exported, you can open it in SIDRA. In this case, we have just exported one site. So, you will see the intersection's geometry and all related information goes to SIDRA. We have the geometry and the phase time. We got the four phases and the yellow time and green time correctly exported. By default, it gives the user-given phase times. You can run it with a practical optimum cycle time, to process the signal and then bring it back to SIDRA again through the signal importation. When it's done, you can save it. And from here you can import all the information. If it is one signal, select the isolated sites button.
- Saif: If you do it as a network, then uncheck this box. Once imported, you will see another control plan has been created for the signal. You can review the system, what was before. You have a base control plan, and then the one that has been imported. All right, here is a quick recap for what we need, but I'm just skipping that. I created an example case. I have this intersection in Great Western highway. I took these four intersections and I exported the whole thing in the SIDRA as the network. So for example, one of the imports has four phases that has been transferred there, and you can see the six seconds interface with the four seconds yellow and 50% red percentage.
- Saif: It gives you four seconds yellow time and two seconds all red time. All the phases have been transferred. Now I want to give some information about how the volume is transferred. You can use the path file, which is the equilibrium, path file to get the turn volume or you can use the turn counts. For roundabouts, the turn counts will give us a good result. I will come back to it later. So the volume, for example here, out of 1532 volume, 374 is taking the left turn, 144 is taking right turn and 1014 is going through. Just showing that the information works well.
- Saif: Either we can use the super node in Aimsun or we can use the volumes from path file. If you use super node that gives you a special turn for those type of roundabouts or complicated intersections. When it is imported back you get the yellow time. It goes here like yellow time for a particular phase, because in SIDRA you can have yellow time changes for any phases I guess. In this test, I have processed the whole network as isolated networks and as a network altogether. We have the original control plans from SCATS, so we can see how the performance of the network changes.
- Saif: If we run it with the SIDRA isolated phases, and with the SIDRA network, we can see that there is a delay time improvement or queue length improvement if you run into the network. Why is that? Various signals are closely spaced. Hence, the coordinator options, will give you a better result. That was the whole approach. Also, you can see that the approach delay for the coordinated approach improves significantly.

- Saif: Now, a few complicated issues we have. As I said, we are still struggling with the multiple nodes. Another issue was the transfer of the solid line or multiple sections that could be used to prevent lane change. We have put it as a barrier in between so that the vehicles don't change lane there. The turn volumes are transferred correctly.
- Saif: Now, the vehicle type: Aimsun can have many vehicle types. We can have car one, car two, car three etc. We can have movement classes in SIDRA, so users may need to do some changes. Now, start loss, Aimsun doesn't use start loss by definition because if you run in microscopic, you have a delay for the signalised intersections which is 1.4 seconds, and also you have an acceleration delay. All together it gives you more than two seconds as a start loss.
- Saif: Similarly, for the meso, there is no separate acceleration delay, but it is combined together, give you more than 1.6 or 1.8 seconds delay. So we didn't use start loss. But we have included end gain in the latest version. So if you're thinking that the end gain will be higher, then you can transfer it. You can transfer the red percentage, and also you can transfer volume from either path assignment or turning. We are always open to suggestions, because this interface is still in development. You can use it if you want to. We have it ready in the latest version. That's all.
- Rahmi: Let's get some questions.
- Mark: I just want to make a comment about the vehicle types. SIDRA provides facilities for up to 12 movement classes. So there's no problem with separate classes of light vehicles and trucks. We could easily have buses, bicycles. We have the standard heavy vehicles class and there's a large truck class. You could introduce user-defined movement classes for very large trucks, taxis and so on. So, just wanted you to be aware.
- Saif: We need to make sure that all the volumes are transferred. You can have demands coming under many vehicle classes. So if you have car one, car two and car three, then you will have traffic demand for car one, car two and car three. If you cannot transfer all of them together, the sectional volume will differ. So you need to make sure that we get all vehicle type. We can change them to vehicle classes and choose that vehicle class to transfer. So all the demands under the same classes can be transferred easily through this interface.
- Rahmi: In fact, our recent experience is that state road authorities in Australia are very keen to model very large vehicles. Even road trains are coming to signals and so on. So they want to include more heavy vehicle classes into the analysis, because they have very different characteristics like huge difference in acceleration characteristics of large heavy vehicle types.
- Delegate: When you demonstrated Aimsun processing SIDRA site, you had some user interaction during that process. Can you explain reasoning behind that or why that was the case?
- Saif: Yes, there are many user interactions. You can have flow from a static model, from microscopic, from mesoscopic. All things come in the same interface in Aimsun. So, we have to make sure which one you are exporting. Also, you may want to export all nodes or individual nodes. We know that there is a limitation of 20, so if your network has two hundred intersections that might not work. Then you have to select a small boundary or select these 20 nodes separately.

- Saif: We could not make it much more automatic than this. I know there is a lot of user interactions and I think it's a good sign, because I mean you have to understand what you're doing. So, the users should do it as step by step. We did not design it as a single key click that process it and come back to us, because we want the users to process it in SIDRA. And there are many other options to do the processing. We don't know what the user would prefer for a particular site.
- Delegate: Yes, they were the more common reasons I was asking about.
- Saif: We want the users to make an informed decision. That's why there are some user interactions.
- Delegate: Is the Aimsun-SIDRA API you're developing available through the Aimsun website? Is it going to be linked to SIDRA documents for downloading and use?
- Saif: The one available in the website at the moment doesn't have this functionality. But if you need it, it's available. We can give you a special version for that. The development is done. The one I have shown is from the final version, so it is readily available.



5 Tracks – SIDRA Link

Grant Smith, Transportation and Traffic Systems Ltd

Rahmi: Grant comes from Christchurch, New Zealand. His program TRACKS has a long history. As old as SIDRA I understand. Okay, go back. He always had some connection to SIDRA even many years ago with the text .DAT files and so on. And some users in New South Wales for example kept using that link, but in recent years, Grant has updated TRACKSlink to the most recent SIDRA using the API. So it's a good experience, and he will present to you what he has done and what he has achieved.

Grant: I just want to give you a very quick history of TRACKS and explain the reasons why we're actually here today. TRACKS started in 1975 when I returned to New Zealand from the UK, and there was no commercial software available to run transportation models. So we had to write our own. And over the 40 something years since then, it's developed into the software that it is today. It's a full four step transportation planning suite including trip generation, starting with land use, and going through distribution, mode split and assignment. In the assignment stage, we can use very large networks such as the Sydney SDM model, which we're running for clients in Illawarra. Or we can go down to very small networks such as the little one here, which the small town in New Zealand.

Grant: The intersections within the assignment are explicitly modelled. We started doing this in 1983. In 1995, we went from approach-based intersection modelling into lane and movement-based modelling. That's when we started working with the SIDRA equations. Currently, the roundabouts are simulated using formula which we have adapted from the SIDRA 5.1 equations. For priority intersections, we are using Tanner's queuing theory which Kimber and Holis modified in 1979. Fisk and Tan modified the TRRL equations in 1989 for a limited number of movement types, and then Gabites Porter extended that work in 1991 as a research project for what is now the New Zealand Transport Authority. We use the TRACKS network editor to code priorities and roundabouts.

We used SIDRA 2.1 in the early stages to code the SIDRA .DAT files. With the demise of Version 2.1, or at least the demise of Windows XP machines, we started having to do that manually. It was fraught with error and we thought we've just got to get into electronically linking with SIDRA so that we're using the SIDRA UI to do our coding.

Grant: Now, I've got two ways that I want to quickly explain to you this afternoon. One is the way TRACKS uses SIDRA, and then secondly, the way SIDRA users can use TRACKS.

Grant: We firstly have to convert the version 2.1 for files into a SIP file, so we have to go through SIDRA 5.1. The same happens with the legacy models, and we've got dozens of them around New Zealand and Australia.

Grant: Now, TRACKS knows enough about roundabouts and priority intersections to be able to code those directly into SIDRA without any further data being required. The priority intersections, as with roundabouts, we can code directly, manually or electronically, but the operation is very, very slow in large networks. For example, a network that we've got covering Camden, Wollondilly, and Campbelltown, that's the Macro area here just south of Sydney, has got 250-something signals in it, about almost 300 roundabouts, and we

are doing a SIDRA analysis on that for every one of those intersections for every increment of loading during the assignment. And for those of you who understand four-step modelling, you'll know what I'm talking about.

Grant: Without using SIDRA, that assignment takes 20 minutes. It's got about 17,000 nodes in it. We weren't prepared to put the priority intersections into it. Using SIDRA, it takes about 40 minutes, roughly doubling the time. What we've now got though, is a seamless interface between SIDRA and TRACKS, so we can run TRACKS, we can alter the design in SIDRA, and then rerun TRACKS.

Grant: So, effectively what we've done is add the assignment process to SIDRA so that we're getting new paths and new volumes every time we change the design. We can now use either TRACKS or SIDRA graphics outputs for reporting. The results are identical. Flows the same, delays the same.

Grant: Essentially, we can run TRACKS the way it was before using the ARR 123 equations, the SIDRA 2.1 files, or we can use SIDRA simply as the database for the signals and roundabouts and just read the data from it. Write flows back if you want to. Or we can use SIDRA to do the processing. And, the models that I've converted, where we used SIDRA to do the processing for signals and roundabouts, we're getting a very much better validation than we were before.

Grant: Making a SIDRA network is quite quick and easy. You cut a sub area in TRACKS and TRACKS writes the site files and the flows into a SIDRA intersection project file from the sub-area that forms the SIDRA network. You can then process the network in SIDRA intersection either as individual sites or as the network, and then use TRACKS to display the flows or use the SIDRA graphics to inspect the best of both.

Grant: Estimating flows, and this is one of the areas where if you have a intersection or particularly a network, we've got some of the intersections counted but you haven't got them all, just use the matrix estimation. You can either get the flows directly out of a TRACKS model or just use the matrix estimation ability in TRACKS to fill in all of the missing flows.

Grant: And then, what we've done now, because transport planning modelling, four step modelling software is expensive, we've put together a limited number of programs and a limited area as a direct complement to SIDRA. Fully functional three-step modelling software, simplified for small networks. We've designed it to be used by traffic engineers with limited modelling experience. It's intended to complement SIDRA and has been developed in very close cooperation, in particular with Harry, to get the two programs doing the same thing. It's very inexpensive, about the same price as SIDRA, and it comes with a complete working model, a step-by-step user manual, and a complete software user manual.

Grant: That'll do. I'm not going to talk about algorithms. I'm not going to talk about code.

Rahmi: Let's have some questions.

Delegate: So, the three-step modelling means you don't do assignment?

- Grant: Oh no. You do generation, if you want to, generation from land use, distribution and assignment. So it's a vehicle model. If you want pedestrians or cyclists or buses, put them into the SIDRA file, because they are fixed route.
- Grant: So what TRACKS is doing is providing the new paths based on whatever changes you've made or whatever delays are coming out of SIDRA.
- Rahmi: So, you said traffic generation is included?
- Grant: Yes. The package was released last month.
- Delegate: It's a simple question really, but I'm just wondering who is using TRACKS and who are your potential customers?
- Grant: At the moment, it has been used in almost every city in New Zealand. It's being used with every local authority from Camden down to the Victorian border in New South Wales. We've got models in Dubbo. We've got models in South Africa and Malaysia.
- Who would use TRACKS_LITE is anyone who is using SIDRA network at the moment and wants to be able to get flows quickly and easily.
- Rahmi: We will continue with presentations. We have two more presentations and then we will go to general discussion as the last session. The next presenter now is Paul van den Bos.



6 Using SIDRA API for EMME and DYNAMEQ

Paul van den Bos, Freelance Transport Modeller

- Paul: Thank you so much. I'm going to talk about how I use SIDRA API. We use both Emme and Dynameq which is an add-on to Emme. Winnipeg is the standard network used in training of Emme. On the left, we have Emme and I've just transferred the network from Dynameq; they are identical of course.
- Paul: What I'm going to do is look at this one intersection in both packages. This is a Google map of it. It's the busiest intersection in Winnipeg. They are driving on the right-hand side of the road. This are views looking north, south, west, and east.
- Paul: When I build the bank, I build it as a left-hand drive bank and I build the SIDRA SIP8 also as left-hand drive, with the New South Wales option, so that we can get the numbers straight out of Emme, straight into Dynameq and SIDRA.
- Paul: The way I use SIDRA in Emme is simple - I do all my network editing in Dynameq and then I transfer the network into Emme. Generally, in Emme, I do all the matrix calculations and then transfer the matrices back to Dynameq so that I have identical models in both software packages, and I should get similar results when I run them.
- Paul: When we start a brand-new project, we need to have volume-delay functions in Emme. I use the Akçelik curves, and I'll just show you what I do. The curves, if you are not familiar with it, come from a spreadsheet that Rahmi has produced. All the factors are there and if you look at this sheet, you can see the equations with all these parameters. It transfers to a very easy speed-flow curve – called volume-delay function in Emme.
- Paul: The volume-delay function has three variables. Free flow speed, capacity, and the J_A factor. It looks like a stock standard curve in Emme. In this worksheet table you can plot the speed-flow curve – which is plotted as a volume-delay function; most of us are familiar with.
- Paul: As modellers, we are interested in this important part of the curve which is where the congestion occurs. Here is an example of what we do with the J_A factor. You can see even though it is a slow speed road - 40 km/hr, I can still drop 15 km per hour by just fiddling with the J_A factor. Pretty powerful stuff so, for calibration, the J_A factor is the one I tend to fiddle with.
- Paul: Now what happens is, while these curves have an approach delay, I often want to put a little turning delay; but especially in the problem areas, I use SIDRA. This is a Python code and here, I work out the angle between the inbound and outbound links from a calculation of the X and Y coordinates of the approach and departure links.

Paul: Here I convert that angle to left-hand turn, straight-through, or right-hand turn movement. This code allocates just a tiny default delay to the left and right movements; I've got a more complex script than this which takes care of major roads and minor roads delays.

Paul: When I build the network, I calculate default delays at each of the turning movements. You can see the turning penalty function – TPF column - that Emme uses; a U-turn as you can see, it is not allowed, it is coded as a “0”. For right-hand turn it coded “3” and for left-hand turn it coded “1”. There is a banned turn – also coded as a “0”. The straight-through movement is coded as “-1”. So you can see it's fairly easy, stock-standard approach when I build the network: I allocate the volume-delay functions as well as the turning-movement delays in Emme. In this example, all movements have the link-delay and in addition, the left-hand-turn has an extra 1 second delay from the tp1, and the right-hand-turn has an extra 3 seconds from tpf3.

Paul: It's in the problem areas where I have need to calibrate and sometimes I need to add some more delays and that is where I use SIDRA.

Paul: In Emme, we've got the Modeller which works with Python; this is its interface. If you look at it closely, I've added the toolbox to demonstrate this for this workshop. There are two tools in it. When the button is pressed in the first tool, it will import the Dynameq network, do the assignment and then export the turning movements. The second one extracts the Dynameq turning volumes.

It is actually quite interesting ... I'm sitting in Emme and run the Dynameq assignment and export the turning volumes. There are some lower level tool boxes containing Python scripts.

Paul: Here are two turning movement files, one is from Emme, and the other file is from Dynameq. Now I import both these files in an Excel spreadsheet from my usual starting point - this 'Start here' sheet. Press this button and it imports the Emme turns. Press that button to import the Dynameq turns.

Paul: You can see here on the left-hand side, Emme has the one hour turning movements, but for Dynameq, because I sliced it in fifteen minutes, I have the numbers represent fifteen-minute hourly flow rates.

Paul: And the numbers in this plot are just simple look-ups.

Paul: I use this for the Emme data input. You can see here, that the layout of the Emme turning volumes is identical to the SIDRA turning movement layout shown when we click the View Display > Volumes. Now I manually enter these Emme numbers into SIDRA. It is a straight transposing so that I don't have any data entry problems.

Paul: Essentially, when we run SIDRA with these Emme volumes, we obtain the correct delays and the green times. Using this output we can make some nominal adjustments to the Emme turning movements. As you can see, my turning penalty file is very simple. The fp1 = 1 second delay, fp2 = 2 second delay, and so on.

When I look at a segment and there is too much traffic ... well it may need a few more seconds delay to make it less attractive on that segment, and vice versa.

- Paul: In Dynameq, it's very different. I'm going to talk about some of the bits that you've seen earlier.
- Paul: There are four phases for this intersection, and a critical item is the green time, which is generated in the SIDRA API, and that value needs to be used in Dynameq. This is the Dynameq control file, and this is the intersection node number and we can see the Link IDs of the different turning movements but I'm only interested in the green times for each of the movements.
- Paul: Here is the overview of Dynameq and SIDRA API. Within the SIDRA API, I populate the turning volumes with my Dynameq outputs. Run the API that generates the green times, which I write out to a simple text file. There is a Python script that updates the control file, runs the DTA and generates a new set of turning movements. I cycle this a couple of times until I get it right.
- Paul: To get the green times, this is of course for all the intersections, notice it's only that bit of data that I'm interested in: the published green time. Because that's the bit I want to use for my intersections. I have direct access to the Dynameq control file, and I just read the green time and update it.
- Paul: Here are the details. You can see here, I have modified the SIDRA API workbook to import the Dynameq link volumes and turning volumes outputs. You can see the SIDRA API sheets.
- Paul: Now, if you look here, this is the Dynameq link sheet, and notice it's got the direction of the link – Northbound – Westbound etc. This is useful for the table in the inputs sheet. But notice this is written for an international market and therefore it has a very fine gradation “south-westbound 2” direction.
- Paul: So, once I import the turning movements in this sheet, we can see I've put in a table that looks very similar to the volume inputs of the API format. And there's a slice for every 15 minutes.
- Paul: And you can see here, the control panel that we've seen earlier on from Mark's presentation, I've added some radio buttons on which slice that I want to copy in. The Yellow button copies the Dynameq turns into the API's Volume sheet, the standard “Run SIDRA” button runs the API, and then I extract the SIDRA green times with this Green button.
- Paul: I got some help from SIDRA people on how to do this; this is the SIDRA API code. You can see the first three modules are the SI modules from SIDRA API people to run the API. All the other modules are mine. You can see the API code that extracts the Green times.
- Paul: I display these green times and write it out to a simple text file. Here's my Python script that picks up that text file. I've got this lower level routine to fix up all the bits inside the Dynameq control file.
- Paul: Right now, Dynameq is able to optimise traffic lights but at only on the hourly basis. I think I'd rather stick with the 15-minute approach - so that's why I want to continue to use it. I have found that in any network, there are local short-sharp congestion issues. For example, near schools, especially if there is a cluster of schools, this peak occurs between 08:30-08:45. If there is a shopping centre as well, the peak may be between

08:45-09:00. The hourly flow it tends to wash out such a peak-within-a-peak issue. The system may take a long time to recover from such a local congestion issue. The reason why I'm here is to learn how we can do it with Python instead of having to back-press all those buttons, I'd rather press one button and get all that stuff done behind the scene. Harry just demonstrated that capability early in this workshop.

Paul: Now what I want to do in the last couple of minutes is actually to give a bit of background to some bits that's I've found.

Paul: The next few slides are very important for us as SIDRA modellers.

Paul: Slide 39 shows the speed-flow curve that comes from the SIDRA manual. It shows the maximum flow, and this oversaturated Region C, represented by this brown curve, that's important. This curve is used in all strategic modelling to determine the "hot spots" in the system. We get the correct speed but the flow is much higher than the capacity. For the study that I was asked to look at, we're talking about volume over capacity ratios of 1.4, 1.6, 1.8, way up there. This knowledge is important for economic analysis, because the economist believe, that all these drivers and passengers are going to get the benefits. From this graph we can see that only a portion of that number will get the benefits because the rest of the traffic cannot fit on the link. Where will that traffic go?

Paul: Here is a typical output from SIDRA. Notice the difference between the totals of the demand and arrival flows. In this case about the arrival flows are 80% of the demand - because the traffic is stuck in upstream queues. And that's because we're picking stuff up from a strategic model where we have flows higher than capacity. All right, essentially, we have a demand that exceeds maximum flow on this queuing graph ... about there. Here is the number that is actually getting through the intersection. Therefore, keep an eye out for those yellow numbers. If you see yellow numbers, have a good look at it - there is a reason that SIDRA has implemented this yellow highlight feature.

Paul: And this is my last bit. We are often asked to validate our models to travel times. Up to maximum flow, we have what SIDRA calls Unsaturated flow, and we call the green zone. We've seen that graph before, and the red curve represents the red zone - SIDRA refers to it as "forced flow" region. It can also be explained by Region C in Slide 39 which represents travel times measured by travel surveys.

Paul: In Slide 42, these are actually travel times from one of the studies. Notice that there is a cluster of travel times with very low travel times, indicating very fast speeds. There is another cluster that has very long travel times, indicating slow speeds. The Green zone give fast speeds and the Red zone gives the slow speeds. For calibration purposes, we are told to average them out. It's meaningless, right? While the survey was carried out on different days, the results are consistent. We are talking about less than half an hour between the "fast" and "slow" travel times.

Paul: Notice that the flow that we have here, in the green zone, is in an uncongested system. Traffic moves right through the system very easily. Here in the red zone, we have a congested system, in the forced flow region, traffic is queued, crawling through the system. In both cases, there's really not much difference between the traffic flows, but significant difference in travel times. When you do your modelling, think about where we are on that curve, in the green zone, in which we can add more traffic. Or in the red

zone, when the system is operating in the forced flow regime. The traffic count itself will not give that information – we need the speed of the traffic as well.

Paul: Thank you so much.

Delegate: Paul, if I understand you correctly, you run the Emme assignment, take the flows out, calculate delays and therefore green times, and put the green times back and then iterate. How do you ensure stability?

Paul: A very good question. Emme actually guarantees stability. Dynameq also guarantees stability. And that's a kind of interesting question.

Paul: One of the difficult things in Dynameq is that it is possible that your demand is so high, because it is often generated in Emme, or another strategic modelling package - where volumes can exceed the network capacity. What happens in Dynameq is that we store the “access” traffic in virtual queues. At least the traffic will get through the system. You can actually plot these virtual queues and you can see these pillars of traffic at different intersections. You can say well there's six intersections causing all the problems in this network, and then at least you've got something to work towards. Whereas in Emme, the problem doesn't occur because in Emme the traffic can exceed capacity and we have seen in Slide 39 that brown curve tapers off towards infinity. But we can plot the volume over capacity that is greater than 1 to show where the problem areas are.

Delegate: Firstly, it's great to see somebody talk about fundamentals. My question is more a philosophical one. Your understanding is obviously quite good about how models work fundamentally. How well is that understood throughout the world?

Paul: Very good question. I have great difficulties in explaining some of this stuff, in fact I was in Rockhampton doing an advanced course in modelling and I was just absolutely stunned at the lack of fundamental knowledge. This idea of concept of capacity was so rigid, I tried to explain that this was a simplified version.

Rahmi: I'll explain something I conceptualised many years ago. A lot of people talk about this fundamental speed-flow diagram. In transport planning packages, you've got the speed-flow curve coming out to the right showing demand values higher than capacity (Region C). This is different from the fundamental speed-flow curve which has the congested conditions represented by the curve coming to left (Region B). The explanation of region B is that this curve represents data measured by detectors on the road. As traffic slows, we get low flow rates because vehicle headways are getting very long. A particular volume can give two different speeds, one is congested and the other is free-flowing. It is not possible to tell which one using the detector data or the fundamental speed-flow curve. This is important in terms of control systems and traffic modelling.

Rahmi: In the speed-flow curve relevant to the transport modelling packages (Region C), the demand flow rates exceeding the capacity can only be counted at the back of the queue. On the freeway, it is the first time a vehicle joins the back of the queue. The detectors cannot give you demand exceeding capacity. So, the travel times in conditions when demand exceeds capacity can only be measured by travelling in your car through the system. The travel time in the congested section is roughly the time between joining the back of queue and the time departing from the queue.

- Rahmi: So, in modelling and in actual systems, distinguishing between the two types of speed-flow curves is very important.
- Delegate: Paul, you're exporting from SIDRA the green times. Why don't you export the delays? SIDRA has got some quite good formula for determining the delays per movement, why don't you go to that level of detail?
- Paul: Very good question. For Dynameq, I need the green time to run the phases and there are things like offsets between the signals and that sort of stuff that we can get out of SIDRA's network. But for Emme, what we have there is some sort of delay function right? I should have made that clearer but, basically what happens is from the green time you can see these people are getting a lot of green time, therefore there must not be too much delay. This other one has got little green time so there must be fewer cars doing it therefore I need to increase the delay. It's a bit of backwards working but basically we tried to replicate SIDRA's results, the way traffic splits.
- Rahmi: Is Dynameq a microsimulation like Aimsun?
- Paul: Yes.
- Rahmi: So, for Dynameq, you are doing like the Aimsun-SIDRA link in using SIDRA timing data in microsimulation.
- Paul: Yes.



7 GTASIDRA - A Process Integrator

Luca Rosati and Rebecca Strachan, GTA Consultants

- Rahmi: Luca Rosati will present some API work he has been doing for GTA Consultants, and Rebecca Strachan will also join him.
- Luca: I am Luca Rosati, GTA consultant from the Brisbane office. I am the National R & D Manager for the company and Rebecca is our SIDRA and SCATS expert. We are presenting something that we are doing in our company with SIDRA V8 API and so I think that Rebecca can start just for showing you the 'why' we are doing this?
- Rebecca: So at GTA Consultants, we're national. We've got a lot of engineers that are working with SIDRA and one of the biggest challenges when you have a lot of people developing models, is to actually ensure quality control. The consultants out there, you'd appreciate that entering data manually can be both time consuming and enables a chance for risks of typos so we've been working, particularly with Luca, and across our office, to look at a standardised pipeline to minimise some of these risks and ability for errors, so we've been looking at also increasing our working performance and enabling some consistency checks through automating some processes and also then, outputs and quality control.
- Rebecca: So, I've been working heavily with Luca who has been developing some platforms for us and expanding on the SIDRA API to help us out so-
- Luca: At GTA we are developing an analytic platform that is a quite complex platform that allow us fusing, different fusing and blending different data sets about transportation data that is open source data mainly, and to suck from them, information on how transport network works. And, we thought that the best place for using the SIDRA API is into this analytic platform so, we have chosen to use the SIDRA API for giving to this platform, the ability to make signal performances and signal evaluations starting from all the data that are entered into this analytic platform.
- Luca: We are choosing to do this starting from the SIDRA API volumes Excel spreadsheet just because I have taught me that I don't like too much Microsoft scripting, VBA and all this kind of things but, I have taught me two things. One, that the Excel spreadsheet is quite friendly to work; everyone has worked with an Excel spreadsheet and it is better accepted from the consultant than a Python program that is a black box that in some way you click, but you don't understand really where . . . and you have to install it on each computer and so on. And the other one, usually put the hands into the codes written from someone else could be difficult but I have to admit that SIDRA did a very good job I found, easily to modify a plug-in somethings into the SIDRA API so, chapeau, congrats about this.
- Luca: What we have done is, starting from this we have customised heavily the SIDRA volume API. We allowed new functionality to it. The first one is to take all the data, all the information that we have on our analytics platform as an input so we don't have to provide input right now to SIDRA. We take this input, we select the site and we select the day and the AM or PM, and we take the data from our analytics platform and we introduce new functionality that allow us to process multiple sites at the same time, or

to make also . . . we have introduced the networks processing on the SIDRA volumes that were not present, if I remember. And we have customised all the reporting.

Luca: So now, with the Excel spreadsheet integrated with the GTA analytics, we click a button ideally, and we have all the reports ready for using them into our report system and so on.

Luca: So, this is how we did it. Simple things. So starting from the Excel spreadsheet but things that works. So, this is what we have done exceptionally-

Rebecca: I'll just jump in. In terms of the inputs, and this is where it comes down to here is, for consultants, we have often got turning movement counts or SCATS detector volumes, so we've basically customised this so we can import directly from any of these formats that we've got using that open source data, so that's where our time savings are so-

Luca: Yeah.

Luca: And so as you can see, it's quite different from the standard Volumes, so we allow the consultant to choose what kind of output he needs because some time when you need an [inaudible] optimization, you can have 10 sites and so you don't need to able the output that now the spreadsheet provides, so something like that. You can generate inputs and optimization for one site, for all the sites that are into the SIDRA model as a [inaudible] or you can just process different network and having the network results or read the ... in terms of tables to copy and paste into the report and this is where the integration with the GT analytics works. So GTA analytics provide the inputs in terms of traffic volume for each detector, each 15 minutes for different periods. And here, we have introduced a combo box for providing the percentage of heavy vehicle, so we take into account also the heavy vehicle. This is a flag that allow us to understand if you want to use the input from our analyst platform, or you want to provide input to by your [inaudible].

Luca: The spreadsheet allow you to ... if you put here three, four hours time period, it can understand what is the peak hour, select the peak hour in this period, but you can also select your peak hour to use to use the same peak hour for different signal for doing network optimization that is consistent in the time period [inaudible 01:46:50]. So that is, we want to do more as usual, I have some problem in understanding and processing the errors that come from SIDRA into the Excel API, because I know if there is an error, but I can't understand what kind of error is, and every time ... probably I don't know, is my fault. If there is an error, every time I have to reopen SIDRA, understand what is there with the Diagnostic. So this is something that we can understand, and what I want to do really is to allow our consultant to use the spreadsheet to customise the SIDRA model more and more, so accessing different parameters and change them. And probably during my spare time and for my pleasure, I would like to make some Python scripting using the SIDRA Python API. So this is how the SIDRA API, the SIDRA Volume starts transforming the way that we work in GTA and thus giving us a very good results in terms of performance and quantity.

Rahmi: One thing we are doing in SIDRA INTERSECTION Version 9 is to break up report elements and give more control to the user. In Version 8, we introduced user reports. In the templates for user reports, you can choose elements of reports. In Version 9, you will be able to control which tables you want in standard reports.

- Delegate: I hope not much will change in API for Version 9.
- Delegate: I've prepared a similar API application for SIDRA INTERSECTION Version 7. I had to prepare it again for Version 8, but all I had to change was the pointers to which API is looking at, it was quite a simple. But ... all the rest of the work book just exists, it's just the bit that does the API changing and almost literally just lift their code and overwrite your sheet and it's almost that simple. With my version, which SIDRA's, which set of volumes, which sites I've got in a row, so you can see them right there and you can fill it out for absolutely every SIDRA want to do in this particular project. Where do you have your list of SIDRA sites volumes?
- Luca: These are the lists of the site if you want to select site, but you can select one by one or generate for all the sites of the input.
- Delegate: So that's something that comes from your model, I pull all that data and list on that sheet. So I've got a sort of endless set of rows underneath there, and you just say, okay, I want all those, and then I've got to tell it to flag where the executor run that this time or not.
- Rebecca: So almost like your tick box if you've got slides in there, you can just select five.
- Delegate: Yes, although sometimes it comes out to be hundreds of sites and I want to run seven.
- Rebecca: Yes, always evolving and you probably ... since you've come from SIDRA 5 you've probably worked out processes that work best for yourself and for your team.
- Luca: Thank you for the idea. It is a good idea to bring in the call.
- Rebecca: The main thing is just to show that they're completely adaptable and you do work out what works out for yourselves and your team.



8 Discussions on user needs and API development

- Rahmi: We have something like 15 minutes more for further discussion. We can take questions, comments, to all presenters or talking to each other. Just throw something to discuss and let's all comment. So this is a comment discussion time. Philip.
- Philip: To SIDRA, and following on from that last presentation. The error return from your API, I've nailed a lot of the errors and I catch those ... some of the errors you get, I can catch in VBA. So I make sure that the inputs aren't going to cause problems, but that doesn't catch all of them. So yes, I'm another vote for saying it would be nice if your API could actually return an error and a list of what the error meant so that my VBA can pass that through instead of, "hang on, my output's wrong, it's failed, I haven't found it".
- Delegate: I found the screen empty when I look for the API error, probably I don't know how to do it.
- Moderator: Yeah, when the process failed, you have a return from the SIDRA process that you go to look up the error and I found it empty.
- Delegate: [inaudible 01:52:41] We catch those errors when we run the file in DYNAMIQ mode. As soon as a program finishes processing, the error is lost. You need to catch the error in a text file, an external file. So this what happens.
- Delegate: Just open a text file in the same script as an external file, and it captures everything there.
- Rahmi: Obviously something is not obvious there and maybe we can provide a bit of support text on the website.
- Mark: The fact at least three people are saying this here, there's an issue. I'd like to follow that up. I think this is something where there is some way to do it but it's not easy, it's not obvious. The other issues that I want to ask though is even if you are finding the error, is it helping you to identify the problem? There's always the question, I got an error but I don't know what that means.
- Delegate: I'll give you an example, one of us, we ran the model. The model wouldn't produce a Lane Summary. One of ours we ran the model through the API, and there was an error which was obviously a fundamental error but there was no output into the Lane Summary, Movement Summary or anything like that to give you an indication. And I thought there was a problem with the API, whereas the reality was I had to open up SIDRA to see what that error was. So it was more just if it could give an error message to actually say, hey, you do have to open the software or tell you what it is would be the ultimate so you know exactly where to go next time.
- Rahmi: But when you run SIDRA for that case to see the error, was there a Lane Summary output then?
- Delegate: No.
- Rahmi: So it won't generate it?

- Delegate: If run SIDRA from the SIDRA user interface, it will complain, you get a popup box there saying, this thing needs fixing. When you run it from the API, you just fail to get output and the API doesn't return saying anything.
- Rahmi: Got it.
- Delegate: My tool runs through to completion, but because we've mis-coded three of the SIDRAs, they fail to run, but my code failed to capture that they fail to run. I haven't found how to get the API to say: "I failed". Whereas if you do something silly like leave a space on the end of your site name, which is fairly easy to do, it will not find the site and will actually stop. So my code will fail, we'll know it failed because it literally came back saying, I give up. Whereas there's some where SIDRA won't go, but it doesn't give up.
- Rahmi: Is that API thing - that space stopping?
- Delegate: I don't know. The name of a site in SIDRA can and in spaces, and it's a valid name. I found that it's difficult to get Excel to keep a space on the end of a string. So if I want my site name listed in Excel and then try and match it against a site in SIDRA that has a space in the end of it, it fails to find it.
- Mark: Clearly, we will play around with this error stuff and see what's wrong.
- Rahmi: We can improve it.
- Grant: In defence of the SIDRA guys, we're coding C++ and we're finding the errors, so it might be a VBA issue.
- Mark: Maybe a VBA issue. Still, we at least should have advice about it.
- Rahmi: Because volumes utility as we call it is very popular, a lot of people are using it and doing things, obviously, you can do a lot of things. So we would put effort to improve this -
- Rahmi: Any other questions? Comments?
- Delegate: API is very good in terms of extracting the results from SIDRA, but when it comes to signal diagrams and signal phase diagrams, that's one thing which it is not able to do and if you're using APIs, especially when we have multiple scenarios tested with different phases, it gives output, but in terms of giving to signal engineers to design these phases into their system, that output can't be extracted and you have to open the SIDRA files.
- Rahmi: So you want the pictures?
- Delegate: Yeah the pictures. It is a form of digital recording like if it is a green, it's one and zero and ... pictures would be excellent.
- Rahmi: Yeah, I think Mark indicated, we had a lot of feedback on that at the moment, only Layout pictures are included through the API that we will try to do more of them.
- Rahmi: So in Version 9, we have efforts relating to some fundamental aspect of output, just keep it mind, if we achieve this, we do want to achieve this which I shall mention. We do modelling by moment class, you know the buses and the bicycles and heavy trucks and so on and so on. But some of the displays are just all of them in classes totals are averages are the worst for any class or whatever. So we want to give you options to select a movement class and look at the display, e.g. displays for buses only, and so on if you want it in your report. This is one of our targets for version 9. It needs going into

some of the fundamental equations, it's not as simple as some calculations need to get more detailed, but we do want that, there's the need there. Yes.

Delegate: How does licencing work with servers? So if I were to install SIDRA on a server and have hundreds of other computers making queries to that, does that attract a different licence, cost licence agreement?

Rahmi: You are talking about general SIDRA licencing?

Delegate: Yes.

Rahmi: Now, we have floating licencing, you are talking about floating licence?

Delegate: I don't know. I'm not IT specifically, so I don't know exactly how licencing works.

Rahmi: We haven't been selling floating licences, like putting SIDRA on a server and a licenced number of users, maximum number of users accessing at any time. We have those around, but they haven't been selling, there's a big trouble with support. Nevertheless, we are really looking at this, we are looking at a bit of change to our licensing system for Version 9 and looking into the possibility of allowing this with any number via a different method.

Rahmi: Harry is working on it and we'll be testing so it maybe that we'll start selling floating licences, especially small to medium size organisations, consultancies, city councils and so on. They just want five people accessing rather than five different PC licences, but there's also importantly, this really very important user group in that category. The current Floating licence doesn't allow for network modelling, there's just up to two intersections, essentially a single intersection modelling. So we are hoping that maybe we can bring network modelling into floating as well. Well, if we achieve this, if it works, we'll be introducing this in version 9. So that's another objective we have got. I hope we can do this.

Delegate: Second question, do you have any mechanism for sharing add-ins and macros and things like forums or a GitHub page or anything like that?

Delegate: For example like AutoCAD has forums where people upload lists of various things that they've built.

Rahmi: You want from Paul what is done already, to run EMME. Well, that crosses my mind.

Mark: One problem I have is that we actually did have SIDRA forums set up for a long time and we discontinued them because no one was using them. It sounded like a great idea at the time, but we weren't getting the activity and we felt therefore our users were probably looking through a different forum or a different methodology, maybe that has changed, technically it is not a difficult thing to do.

Rahmi: But I think something like a LinkedIn SIDRA users group. If someone volunteers to look after it, a control group, that sort of thing.

Mark: Can I just go back briefly to your first question? I just want to pose the question to Harry. Are there any issues related to using the API with a floating licence? Because this is a question that's asked.

Harry: No problem at all. The API only checks whether the software is licensed, it doesn't care about whether it's floating or enterprise or one PC.

- Rahmi: So it's like individual user when they're using ... they want to use API, is it licensed?
- Mark: Any other questions or discussions?
- Delegate: In the original presentation, there was some brief discussion on what the API can do. Are there any limitations specifically to what it can do? So I've gone a bit in VBA based with the API, I'm just wondering, could you take it to the point where you can basically run SIDRA set up sites, do the whole lot in Excel? I don't know if I can actually do that, but I know there are some limitations, is there anything specific that you can think of that would be common day to day that you just can't do it easier?
- Rahmi: Like if you use SIDRA directly, it can do but not through the API.
- Delegate: No so much the graphical issues. I was just asking if you can build an entire intersection outside of the program?
- Rahmi: We mentioned the picture limitation, like as picture output you can only get Layout at the moment, Harry, please tell us, for example building a new site and cloning and importing, all the functionality that you can and cannot do.
- Harry: The import or the clone? Clone, I think I implement that in API, and import, not sure. Maybe it's not there, but main thing is about the diagrams, the picture and things.
- Rahmi: So import, not available.
- Harry: I need to double check.
- Rahmi: Clone, can you clone an existing site?
- Harry: I remember, I have done that – it should be under "project", the method under "project". One thing with VBA, I know it doesn't work well with catching an exception. I remember in VBA, we need to say "on error", "then go to " which line, give a number, very old-style but I think that will improve.
- Rahmi: Okay, we will keep an eye. I think we should make an effort to list very clearly at each stage what you cannot do through the API. Import is more relevant to the user interface really but clone maybe like you already have a Site maybe programmatically you want to clone, change something, clone, change something ... a few geometric parameters and then load the volumes and create scenarios if you like, clone comes in like Network, Site cloning, let's check that one. Import is it really relevant?
- Mark: You might set up a standard project
- Rahmi: Yeah, creating a new project, you want to pick up something from another project.
- Harry: I seem to remember I have an import Site function to take a list of name of the sites and the project file path and the Site names are separated by semicolons. I remember I've done something like that, should be like that maybe. And then another thing is clone network with Sites, or clone networks. Those functions are also available under "project".
- Rahmi: That is great. So, if you have a network set up, say it 10 Sites. In the API, you are doing something with it but yet you want to make exact copy of it including sites. You can make a copy of the network, clone and just change the cycle times. Everything else is the same, but if you clone with the sites then you can go and create a new scenario like more

lanes, or slip lanes, et cetera. They're Site changes, so he's saying you can even do that with API. So that's good.

Rahmi: We'll take the last question.

Delegate: Lucky last. You mentioned before you Mark about using the database as well as the API, I just want you to talk a bit more about why you'd access the database rather than the API?

Mark: Sure, that's a good question. I think one thing might be processing speed, I believe it would be a lot quicker, particularly if you just want to jump in and change a set of volumes, plug them straight into the project file. I believe that's a lot quicker than using the API. You can imagine if you were doing batching through 15-minute volumes for a lot of sites, for the whole day. It might make quite a difference the processing time. There are some speed overheads, just to extend from that question, I probably should have asked Harry, straight after his presentation. I got the impression - maybe it is just an impression - that when Harry ran his little demo in Python, that that was a lot faster than my little demo in Excel. Is that just my imagination or does somehow Python code work faster than VBA? I mean, they're both using COM, and then is that faster than or slower than accessing the database directly? And maybe Grant might have something to say on this, but I'll give it to Harry first.

Harry: I assume maybe Excel is doing something strange to slow you down, the VBA things.

Harry: Direct access to the database will be fast - the fastest way. The API is developed in C#, .NET Framework and exposed as COM, another layer, and then VBA is another layer, and Python is also another layer on the COM, and then COM goes to the .NET Framework and then goes to the database. So it's several layers. If you directly access the database, it will be very fast.

Rahmi: What about Mark said in his presentation that danger with database is that if you are really good, you can even set up a new intersection but what happens in the SIDRA user interface. Say you get a new default intersection, it has four legs. You go to the first dialog, intersection, you delete the North leg because you want a T-intersection. It will automatically go through all other dialogs and change the movements like certain turning moments that don't exist anymore et cetera, et cetera. Pedestrian moments, a lot of data change. So all that change is saved in the database. So if you go into databases that I found this leg, and I'm going to delete it, it will crash the data. It won't work. If you're only doing just volume loading, that's fine, you set your intersection. Say you want to run 200 different volume sets for 10 years every peak a.m, p.m., off-peak, ten of each.

Rahmi: In the future, again, we'll go to variable demand modelling like flow profiling; 15-minute, one after the other for three hours in each case and so on. So all this will be more and more demanding in terms of volume input. So it will help, the normal API helps you to form intersections just the same as the user interface, but the database could be useful. There may be efficiencies in the new database that we will be adopting for Version 9.

Mark: We were thinking it would be a bit faster.

Harry: Saving the data is faster, much faster but reading, depends on what model you are using. If using Entity Framework which is from .NET Framework, it is slower than LinQ direct to the SQL, but saving data into a database file, will be much faster in SQLite. SQLite can

support more data. Now, we have a problem in Version 8 and previous versions if you have more than 200 sites, and if you convert from the old version to the new version, sometimes it crashes. That's something about the SQL Compact Server problem, and now we've changed to the SQLite, we don't have that problem anymore.

Rahmi: Okay, I think that is it, that was the last question. Thank you very much for your interest in this workshop. In some ways, this became like a user group meeting. We've done our first user SIDRA user group meeting in Norway this year. We hope to organise user group meetings in Australia during 2019. So thanks for your interest, thank you for coming, especially those from interstate and even from other countries. Now, we will have a very quick evaluation of the workshop, very simple ratings and maybe very quick comments, but please, do let us know if you have other comments any time by email or using our support system.

Rahmi: And another thank you to all the presenters for their contributions to this workshop.

