Virtual IMS user group presentation

The latest webinar from the Virtual IMS user group was entitled, “IMS and APIs (the Good, the Bad, and the UGLY!)”. It was presented by Dusty Rivers, Director, z Systems Software: IMS & CICS at GT Software.

Dusty Rivers has an extensive 39-year background in global mainframe systems integration, and has been recognized as an IBM Champion for Information Management for nine years in a row for his leadership and contributions to the Information Management community. His expertise focuses on enabling global organizations to extend the use of their mainframe systems into the world of Web services, clouds, and all distributed systems. An engaging speaker, Dusty travels the United States presenting on various topics in leveraging IMS for modern technologies.

Dusty started by looking at what was meant by an Application Programming Interface (API). Wikipedia tells us that an API is a set of subroutine definitions, protocols, and tools for building application software. In general terms, it is a set of clearly defined methods of communication between various software components. A good API makes it easier to develop a computer program by providing all the building blocks, which are then put together by the programmer.

Contents:

- Virtual IMS user group presentation 1
- Meeting dates 5
- Recent IMS articles 6
- Arcati Mainframe Yearbook 2018 6
- About the Virtual IMS user group 6
An API may be for a Web-based system, operating system, database system, computer hardware, or software library.

An API specification can take many forms, but often includes specifications for routines, data structures, object classes, variables, or remote calls. POSIX, Microsoft Windows API, the C++ Standard Template Library, and Java APIs are examples of different forms of APIs.

Documentation for the API is usually provided to facilitate usage. And they all look different (see Figure 1)!

Many people don’t even think that IMS can be part of the bigger computing infrastructure and view it as a separate island of computing with its own special way of working, eg Db2, CICS, IMS, etc. Figure 2 shows that all
those things we associate with the mainframe can have APIs and can share data using the techniques familiar to people working with mobile devices.

Figure 3 illustrates the granularity of APIs. Atomic APIs simple move one single piece of information, eg a customer’s credit rating. Whereas composite APIs can move large amounts of information, eg all of a customer’s details. And this can be passed along the application.

Dusty then went on to look at actual APIs, which he categorized as:

- COBOL and PL/1
- IMS transactions
- IMS transactions combined
- IMS conversational
- IMS multi-segment output.

<table>
<thead>
<tr>
<th>THE GOOD</th>
<th>THE BAD</th>
<th>THE UGLY</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Data Structures Supported</td>
<td>Some structures don’t map well to distributed Apps</td>
<td>Comp-3, Binary , ODO REdefines</td>
</tr>
<tr>
<td>All can be exposed as service inputs/outputs</td>
<td>Names in COBOL-PL/1 may be cryptic and need to be renamed</td>
<td>c3-7-proj1-open-partnum12 = partNUM</td>
</tr>
<tr>
<td>Can expose existing programs without changes</td>
<td>May need more data to drive than the app knows</td>
<td>Message switches, and other calls</td>
</tr>
</tbody>
</table>

Figure 4: The good, the bad, and the ugly for COBOL and PL/1
<table>
<thead>
<tr>
<th>THE GOOD</th>
<th>THE BAD</th>
<th>THE UGLY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Existing Transactions can be exposed as REST or SOAP</td>
<td>A Transaction may be too fine grained</td>
<td>Multiple Transactions may have to be used in service</td>
</tr>
<tr>
<td>Data from transaction returned as a service output</td>
<td>Data me be to convoluted to use in service</td>
<td>Volume of data may be too large to return to distributed client</td>
</tr>
<tr>
<td>PFKEY = TRANCODE</td>
<td>Maybe need multiple Trans</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5: The good, the bad, and the ugly for IMS transactions

<table>
<thead>
<tr>
<th>THE GOOD</th>
<th>THE BAD</th>
<th>THE UGLY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Combine Transactions in one service</td>
<td>May not work well with others</td>
<td>API’s that run for minutes</td>
</tr>
<tr>
<td>Use Conversational Transactions</td>
<td>Long running conversations may be long running API’s</td>
<td>No understanding of conversational impact</td>
</tr>
<tr>
<td>No Code re-write</td>
<td>May be easier to combine logic to keep from calling multiples transactions</td>
<td></td>
</tr>
</tbody>
</table>

Figure 6: The good, the bad, and the ugly for IMS transactions combined
<table>
<thead>
<tr>
<th>THE GOOD</th>
<th>THE BAD</th>
<th>THE UGLY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wrap a conversation in a service</td>
<td>Wrap a conversation in a service</td>
<td>Wrap a conversation in a service</td>
</tr>
<tr>
<td>Use Conversational Transactions</td>
<td>Long running conversations may be long running API’s</td>
<td>Conversational rollback</td>
</tr>
</tbody>
</table>

Figure 7: The good, the bad, and the ugly for IMS conversational

<table>
<thead>
<tr>
<th>THE GOOD</th>
<th>THE BAD</th>
<th>THE UGLY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiple Segment output can be returned to client</td>
<td>May be variable Length in one response</td>
<td>May be variable length multi-segment response</td>
</tr>
</tbody>
</table>

Figure 8: The good, the bad, and the ugly for IMS multi-segment output

The good, the bad, and the ugly for COBOL and PL/1 is shown in Figure 4.

The good, the bad, and the ugly for IMS transactions is shown in Figure 5.

The good, the bad, and the ugly for IMS transactions combined is shown in Figure 6.

The good, the bad, and the ugly for IMS conversational is shown in Figure 7.

The good, the bad, and the ugly for IMS multi-segment output is shown in Figure 8.

Dusty illustrated a lot of examples using GT Ivory Orchestration, as well as Ivory generated SOAP and REST APIs.


You can see and hear the whole user group meeting by downloading the WMV file from www.fundi.com/virtualims/presentations/2018-02-06meeting.wmv.

Meeting dates

- On 10 April 2018, Compuware’s Thomas Esser will be discussing, “Adding additional IMS Test Environment’s with COPE”.

Virtual IMS user group
• The following meeting will be on 12 June 2018, when IBM will be looking at IMS and Digital Transformation.

Recent IMS articles


*DevOps and IMS: Getting Started* by Saar Sagir on z Systems Developer Community (20 February 2018). You can find the article at https://developer.ibm.com/zsystems/2018/02/20/devops-ims-getting-started/


*IMS documentation is now available in a locally hosted IBM Knowledge Center for easy access* by Emily Siu on z Systems Developer Community (11 January 2018). You can find the article at https://developer.ibm.com/zsystems/2018/01/11/ims-documentation-is-now-available-in-a-locally-hosted-ibm-knowledge-center-for-easy-access/

**Arcati Mainframe Yearbook 2018**

Many members of the Virtual IMS user group contributed to this year’s user survey in the Arcati Mainframe Yearbook.

If you want to get your hand on this one-stop shop for mainframers, the Yearbook is free to download from the Arcati Web site here: https://www.arcati.com/newyearbook18.

About the Virtual IMS user group

The Virtual IMS user group was established as a way for individuals using IBM’s IMS hierarchical database and transaction processing systems to exchange information, learn new techniques, and advance their skills with the product.

The Web site at www.fundi.com/virtualims provides a central point for coordinating periodic meetings (which contain technically-oriented topics presented in a webinar format), and provides articles, discussions, links, and other resources of interest to IBM IMS practitioners. Anyone with an interest in IMS is welcome to join the Virtual IMS user group and share in the knowledge exchange.

To share ideas, and for further information, contact trevor@itech-ed.com.

The Virtual IMS user group is free to its members.