APGEN scheduling: 15 years of Experience in Planning Automation

Pierre F Maldague, Steve Wissler, Matthew Lenda, Daniel Finnerty

Jet Propulsion Laboratory,

California Institute of Technology

Outline

- Introduction to APGEN
- Resource modeling in early versions of APGEN and SEQGEN
- Evolution of the APGEN adaptation language
- Application of APGEN scheduling to space missions
  - Generic morphology of an APGEN adaptation
  - DI/EPOXI
  - Europa (proposed mission)
  - INSIGHT
  - Juno
  - MRO
  - MSL
- Conclusion

Introduction to APGEN

- APGEN = Activity Plan GENerator
- Initially: an interactive tool for the use of mission planners
- As a multi mission tool, APGEN needs to be adapted to each specific mission
- "Adapting" APGEN means "writing a file in the APGEN adaptation language (DSL – Domain-Specific Language)"
- The APGEN DSL describes resources (states that vary as a function of time), activities (blocks of time that influence resources) and constraints (relations between activities and resources that must hold true)
- The APGEN DSL evolved over time, first to provide high-fidelity simulations, later to include scheduling algorithms
- Today, APGEN is used for complex simulation and scheduling tasks both for established missions and for missions in an early development stage

Outline

- Introduction to APGEN
- Resource modeling in early versions of APGEN and SEQGEN
- Evolution of the APGEN adaptation language
- Application of APGEN scheduling to space missions
  - Generic morphology of an APGEN adaptation
  - DI/EPOXI
  - Europa (proposed mission)
  - INSIGHT
  - Juno
  - MRO
  - MSL
- Conclusion

Resource modeling in early versions of APGEN and SEQGEN

- APGEN’s predecessor is SEQGEN (SEQuence GENerator), a multi-mission application for simulating and validating sequences of S/C commands
- The APGEN DSL was strongly influenced by the SEQGEN DSL
- Conceptually, early versions of APGEN and SEQGEN were clean and simple
- Some of that simplicity got lost as more features were added over the last 15 to 20 years

Expansion and modeling stages in early SEQGEN

Stage	Description	Final State
Expansion	Activities are expanded into other activities and commands	All commands are inserted into a time-ordered event queue
Modeling	Commands are scanned in time order and their effect on the model (S/C and ground states) is evaluated one event at a time	All changes in S/C and ground states are recorded in state histories. All activities, commands and resulting state changes are captured in a PEF (Predicted Events File)

Decomposition and Modeling Phases in early APGEN

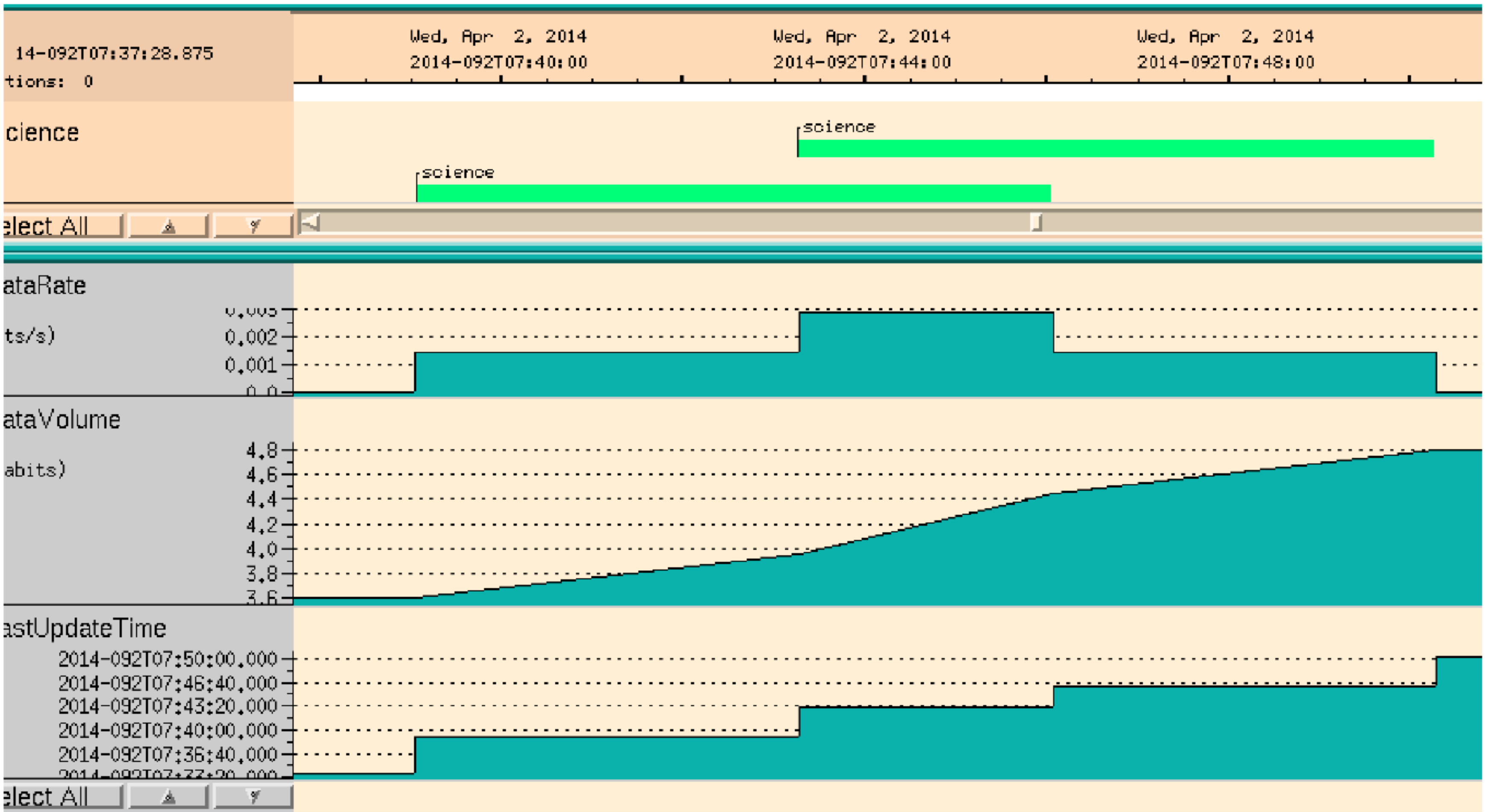
Stage	Description	Final State
Event Generation	Activities generate usage events	Usage events are collected in the event queue
Event Modeling	Events are scanned in time order and their effect on S/C and ground states is evaluated	All changes to the S/C and ground states are recorder in state histories. All activities and state changes are collected in a Time-Order Listing file

Note: APGEN simplifies the SEQGEN setup by eliminating S/C commands. In APGEN, activities can influence states directly through a few pre-defined commands such as "use" and "set".

Non trivial example: science activity (1)

- In spite of the restrictions on it – no iteration, no local variables, limited availability of conditional statements – the early APGEN DSL is surprisingly expressive
- The simple behavior provided by the APGEN *use* and *set* statements leads to abrupt changes in the value of a resource
- Can we change it so that the change in the resource becomes gradual?

Non trivial example: science activity (2)



Outline

- Introduction to APGEN
- Resource modeling in early versions of APGEN and SEQGEN
- **Evolution of the APGEN adaptation language**
- Application of APGEN scheduling to space missions
  - Generic morphology of an APGEN adaptation
  - DI/EPOXI
  - Europa (proposed mission)
  - INSIGHT
  - Juno
  - MRO
  - MSL
- Conclusion

Evolution of the APGEN DSL

- Introduction of standard C-like instructions
  - declarations, function calls, assignments, …
  - data containers (lists, structures)
- Non-exclusive activity decomposition
  - see paper for details
- Introduce concurrent modeling style for better access to multi-threading infrastructure
- Retain old-style, *a priori* modeling style
- Introduction of a user-defined library for integrating external modeling tools
- Addition of a scheduling capability
- See backup slides for more details

Feedback on the APGEN DSL

- Mission Planners and Systems Engineers who have adapted APGEN find the learning curve pretty steep
- The main difficulty is that the DSL controls a multi-threaded modeling environment via a 2-step process:
  - activities and abstract resources create events
  - events modify resources
- This process is left over from the early SEQGEN / APGEN modeling
- The resulting hybrid is not easy to grasp but provides powerful capabilities

Outline

- Introduction to APGEN
- Resource modeling in early versions of APGEN and SEQGEN
- Evolution of the APGEN adaptation language
- Application of APGEN scheduling to space missions
  - Generic morphology of an APGEN adaptation
  - DI/EPOXI
  - Europa (proposed mission)
  - INSIGHT
  - Juno
  - MRO
  - MSL
- Conclusion

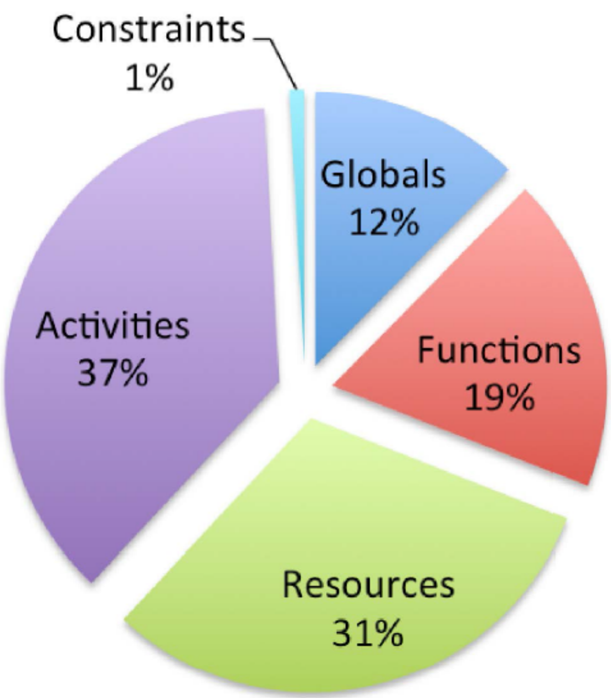
Applications of APGEN scheduling

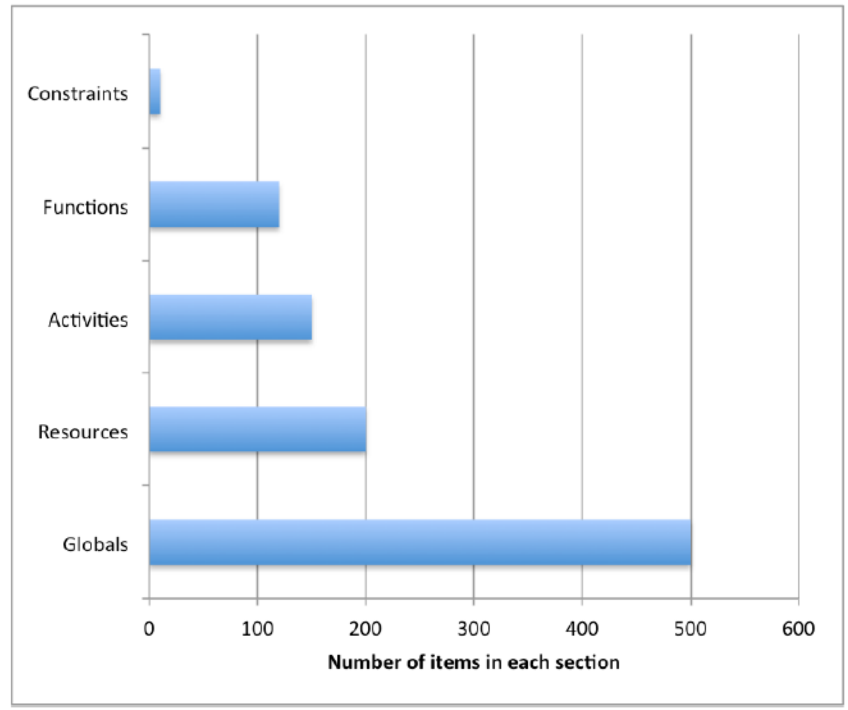
- Ability of APGEN to simulate complex missions and handle difficult scheduling problems has been documented (Ref. 1 of our paper)
- Our focus: Manageability of APGEN adaptation, rather than functionality
- In particular:
  - How much work is required to adapt APGEN to a new mission?
  - How difficult is it to reuse an existing adaptation?
  - How difficult is it to train APGEN adapters?
  - Can APGEN be (partially) replaced by more standard tools?

Generic Morphology of an APGEN Adaptation

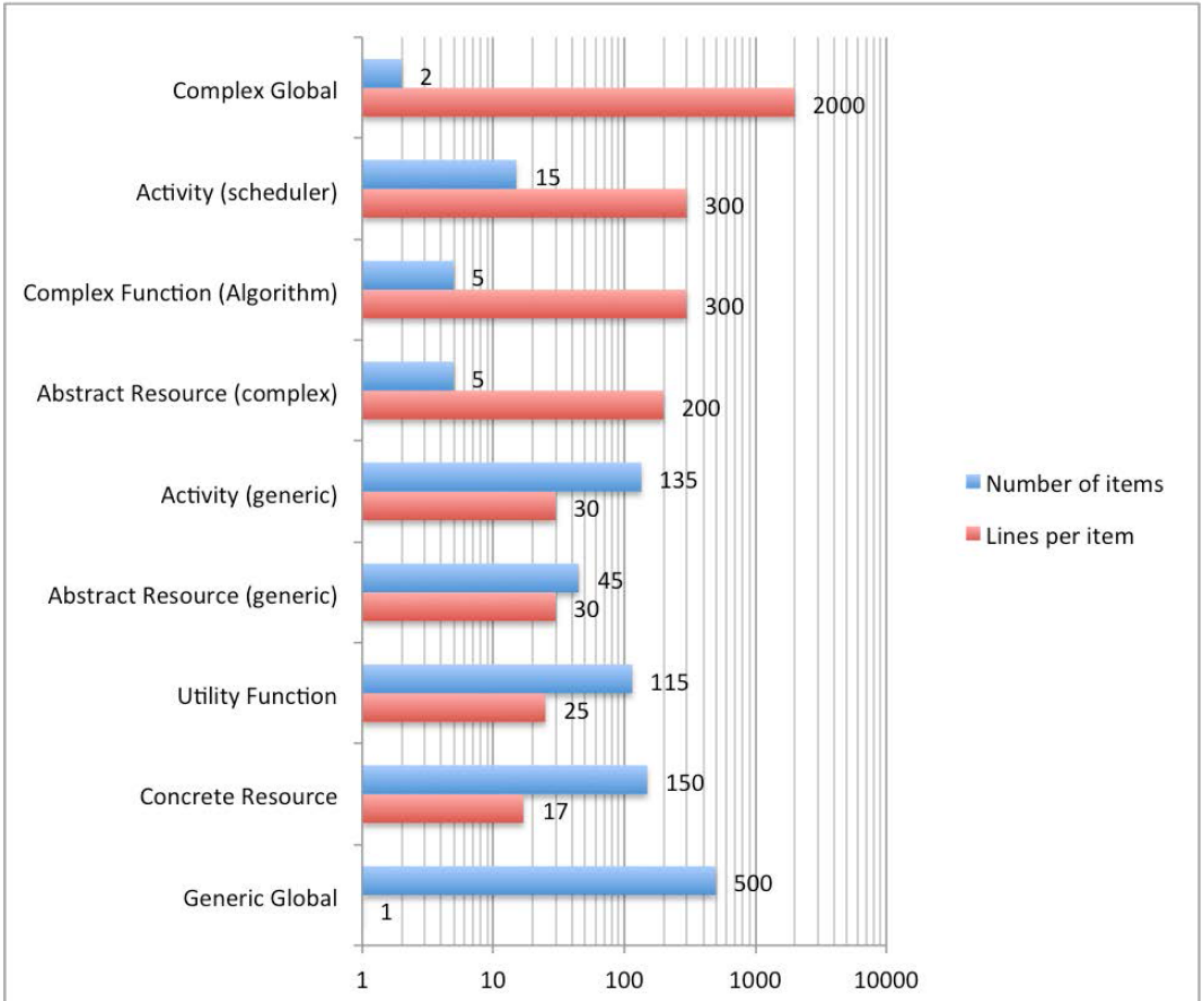
Section	Description
Header	Identifies the file as an AAF (Apgen Adaptation File)
Preliminaries	Definition of mission-specific data types and attributes
Globals	Provides constants and variables that can be accessed from anywhere
Functions	Similar to C functions
Resources	Express S/C and ground states; can be grouped into resource arrays
Abstract Resources	Similar to functions, but can initiate a new modeling thread
Activity Definitions	The heart of the APGEN model; connect activities to resources
Constraints	Diagnostic only – violations are reported but not fixed

What's in an adaptation? Statistics (1)





What's in an adaptation? Statistics (2)



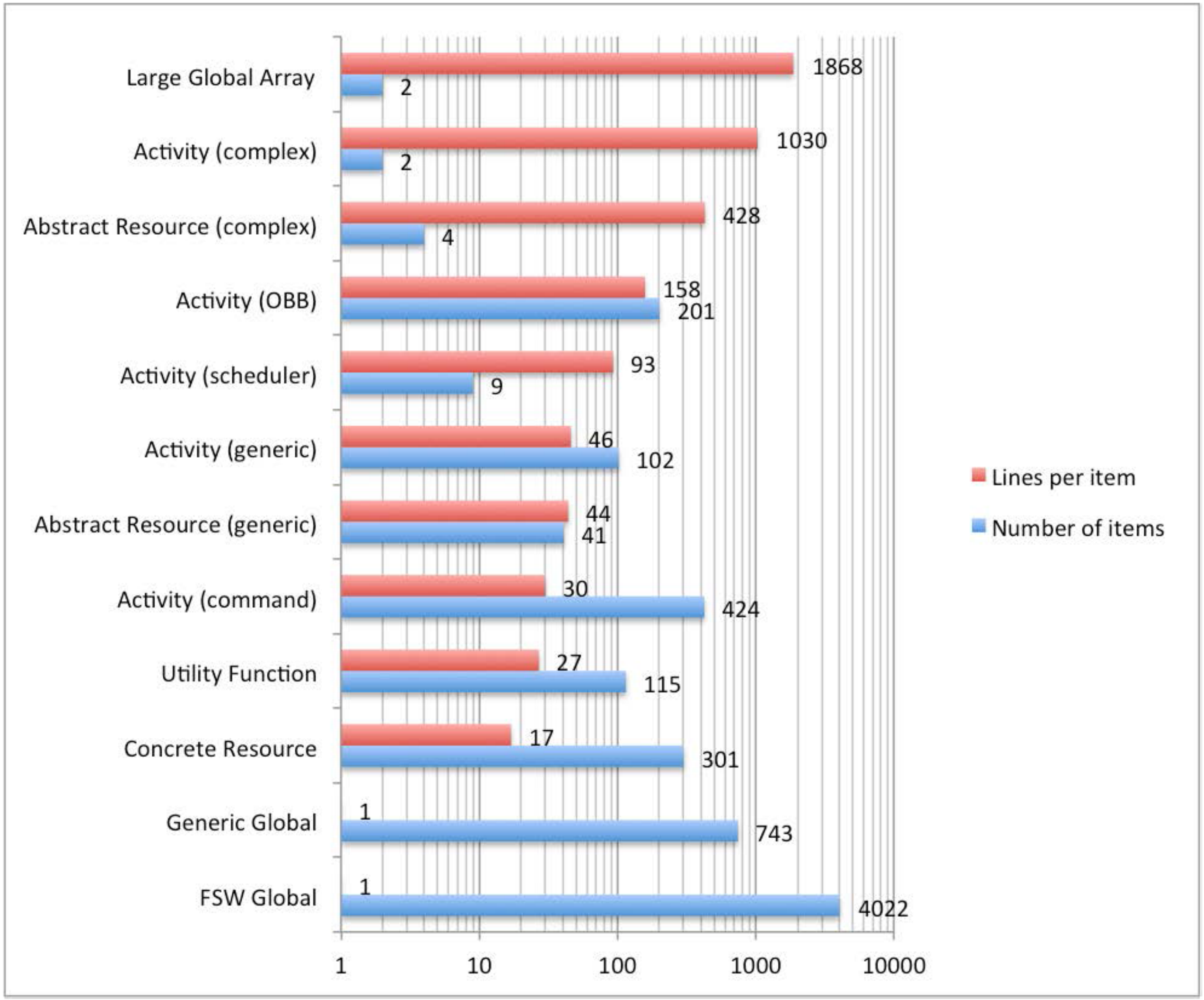
What do adapters do? Some insight

- Most complex objects:
  - complex functions
  - abstract resources
  - schedulers
- Global variables are good candidates for automatic generation
- Utility functions are good candidates for sharing
- Short activity and resource definitions are good candidates for extraction from Systems Engineering databases

Outline

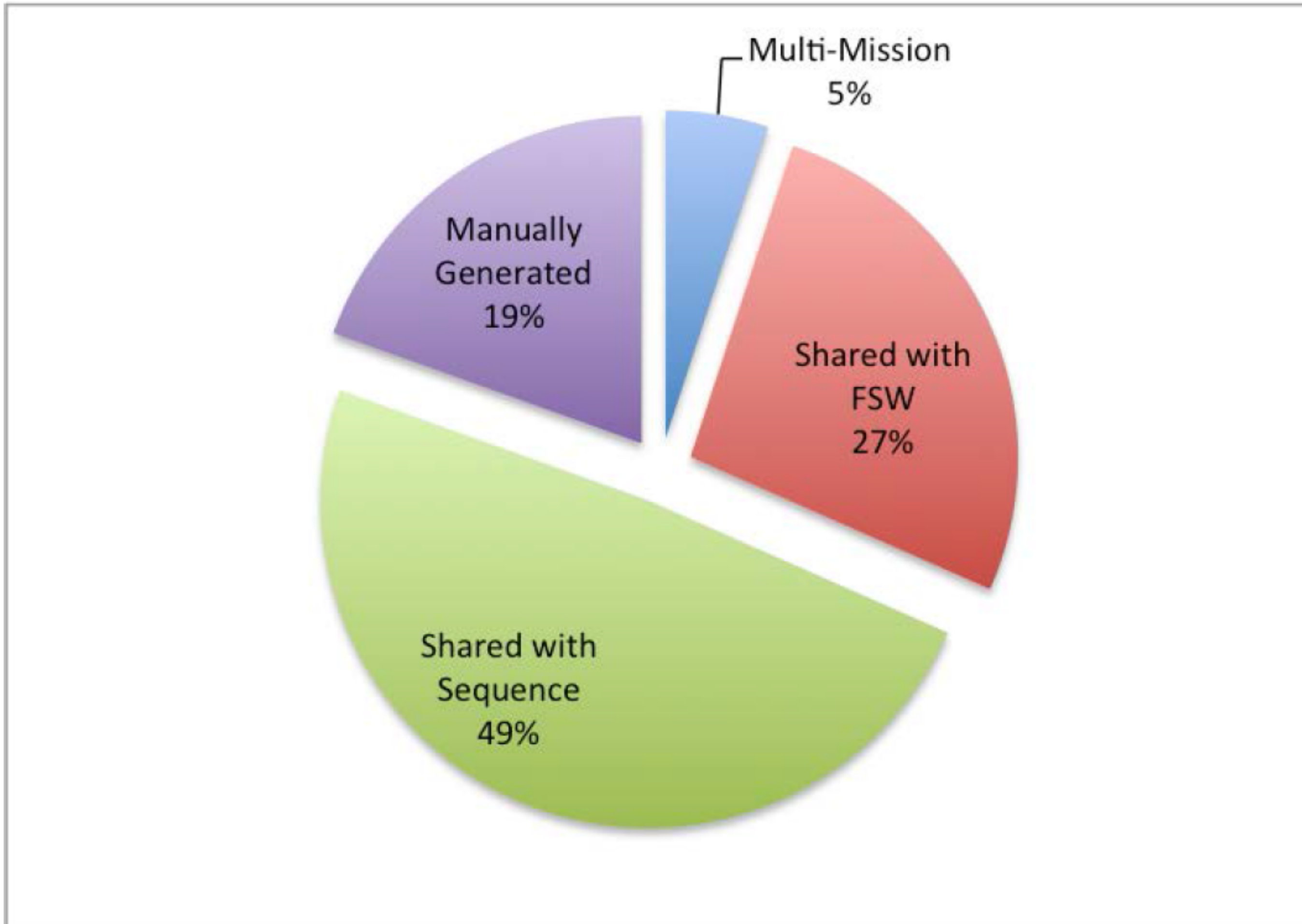
- Introduction to APGEN
- Resource modeling in early versions of APGEN and SEQGEN
- Evolution of the APGEN adaptation language
- Application of APGEN scheduling to space missions
  - Generic morphology of an APGEN adaptation
  - DI/EPOXI
  - Europa (proposed mission)
  - INSIGHT
  - Juno
  - MRO
  - MSL
- Conclusion

DI / EPOXI

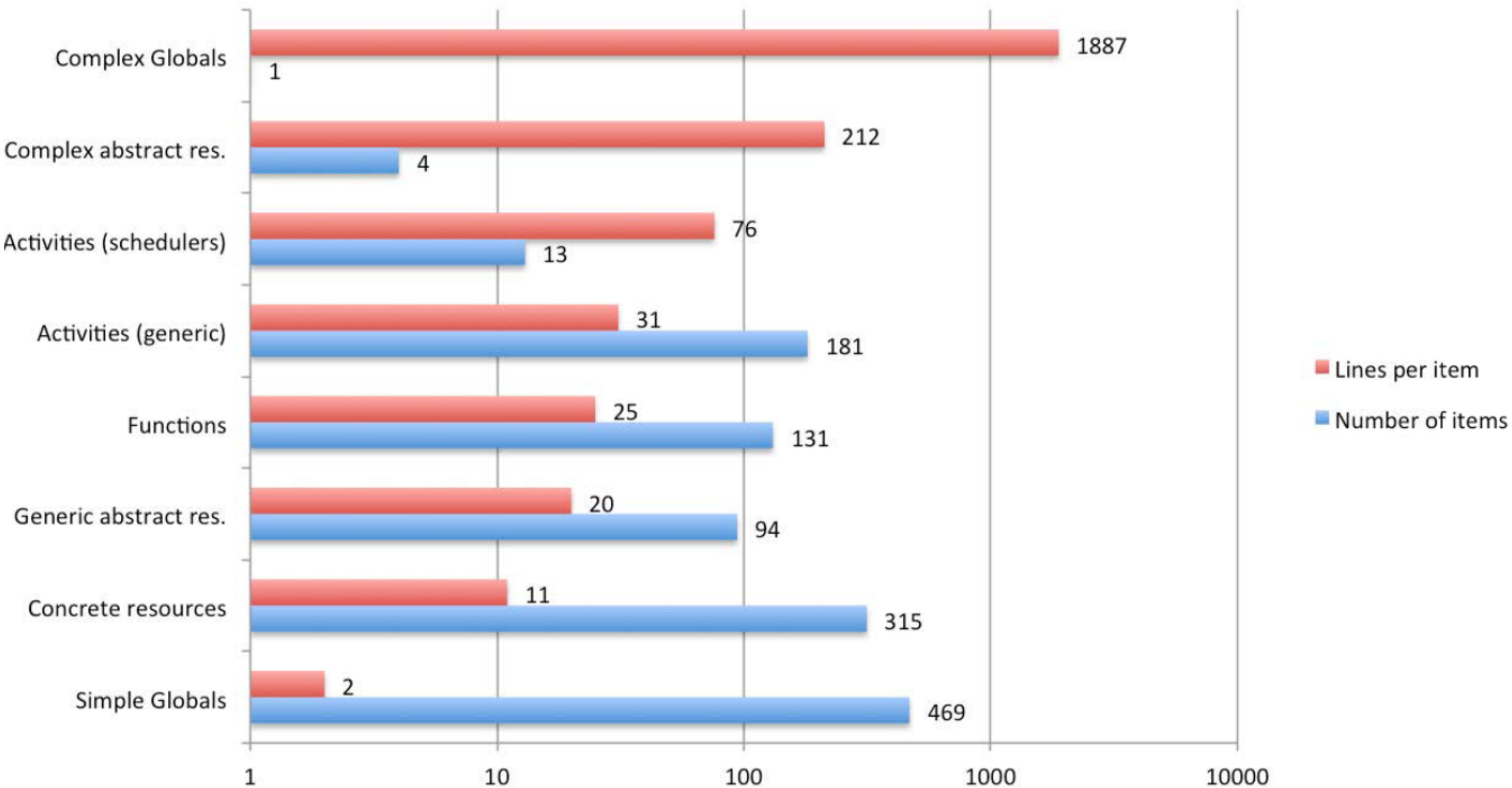


Note: OBB = On-Board Block, a sequence pre-stored in S/C memory

DI / EPOXI Adaptation Characteristics



Europa (proposed mission)



Europa (proposed mission)

- There was considerable reuse of code: among the 131 functions in the adaptation, 82 are common with at least one of the DI, MRO and Juno adaptation.
- About half of the global definitions are concerned with PEL (Power Equipment List) items which were extracted from the System Engineering database by automated methods.
- The Ground model (which includes the DSN model) and much of the Telecommunications model was mostly inherited from DI also.

INSIGHT

- INSIGHT = Interior Exploration using Seismic Investigations, Geodesy and Heat Transport
- “Re-fly” of Phoenix (2008) with different payload
- APGEN adaptation is currently in progress
- APGEN adaptation will be a hybrid
  - power and data models will be adapted from Phoenix
  - INSIGHT instrument activities will be largely autonomous, offering opportunities for automated scheduling

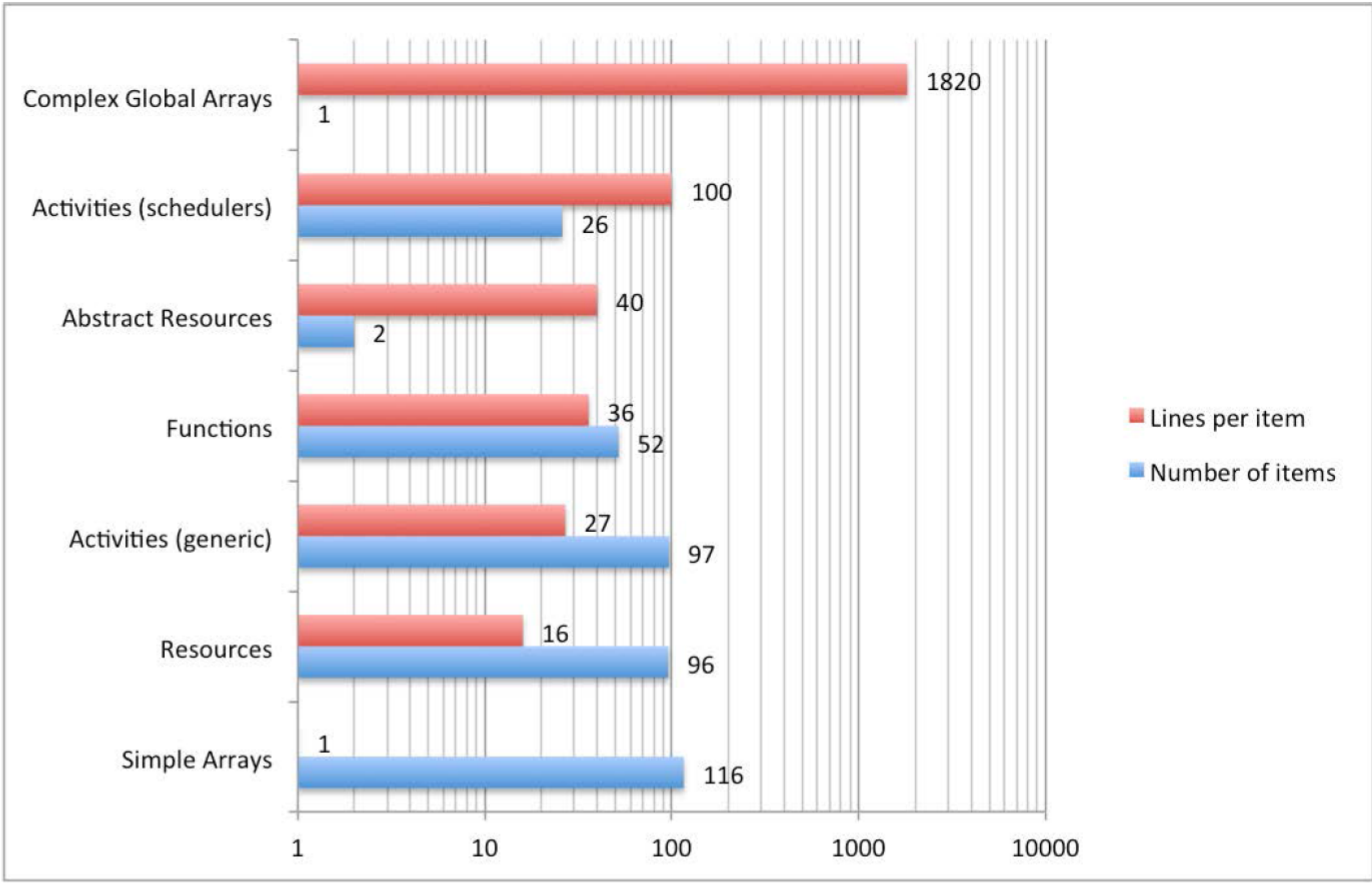
Juno

- Juno is one of three missions using an early APGEN adaptation, Roy Gladden's AUTOGEN
- Odyssey and MRO (Mars Reconnaissance Orbiter) are the other two
- AUTOGEN was enhanced by Matt Lenda to add repointing activities
- Repointing activity scheduling is currently performed in the Cruise phase
- Repointing activity scheduling is currently planned for the Orbital phase also

Juno auto scheduling capabilities

Activity
Automatic scheduling of FSW diagnostics activities
Schedule FINCON requests in the output sequence file
Change the downlink rate on the S/C corresponding to the setup of the allocated dish on the ground
Schedule telecom setup and teardown activities around gaps in DSN coverage
Identify overlaps in DSN allocations and distinguish between handovers and UL transfers
Find all UL transfer times during allocation overlaps to determine additional retransmit times
Identify all 1-way to 2/3-way Doppler Mode transitions after the start of a comm block and manage the DPT (Data Priority Table) mode
Identify all downlink rate changes and determine the DPT/FPT (Frame Priority Mode) and comm_begin/end management commanding
Query resources and states at the sequence cutoff time and print them to an output file
Identify and define parameters of repointing activities during the Cruise phase
Schedule repointing activities to slew S/C's HGA boresight at the Earth during the Cruise phase

Juno statistics



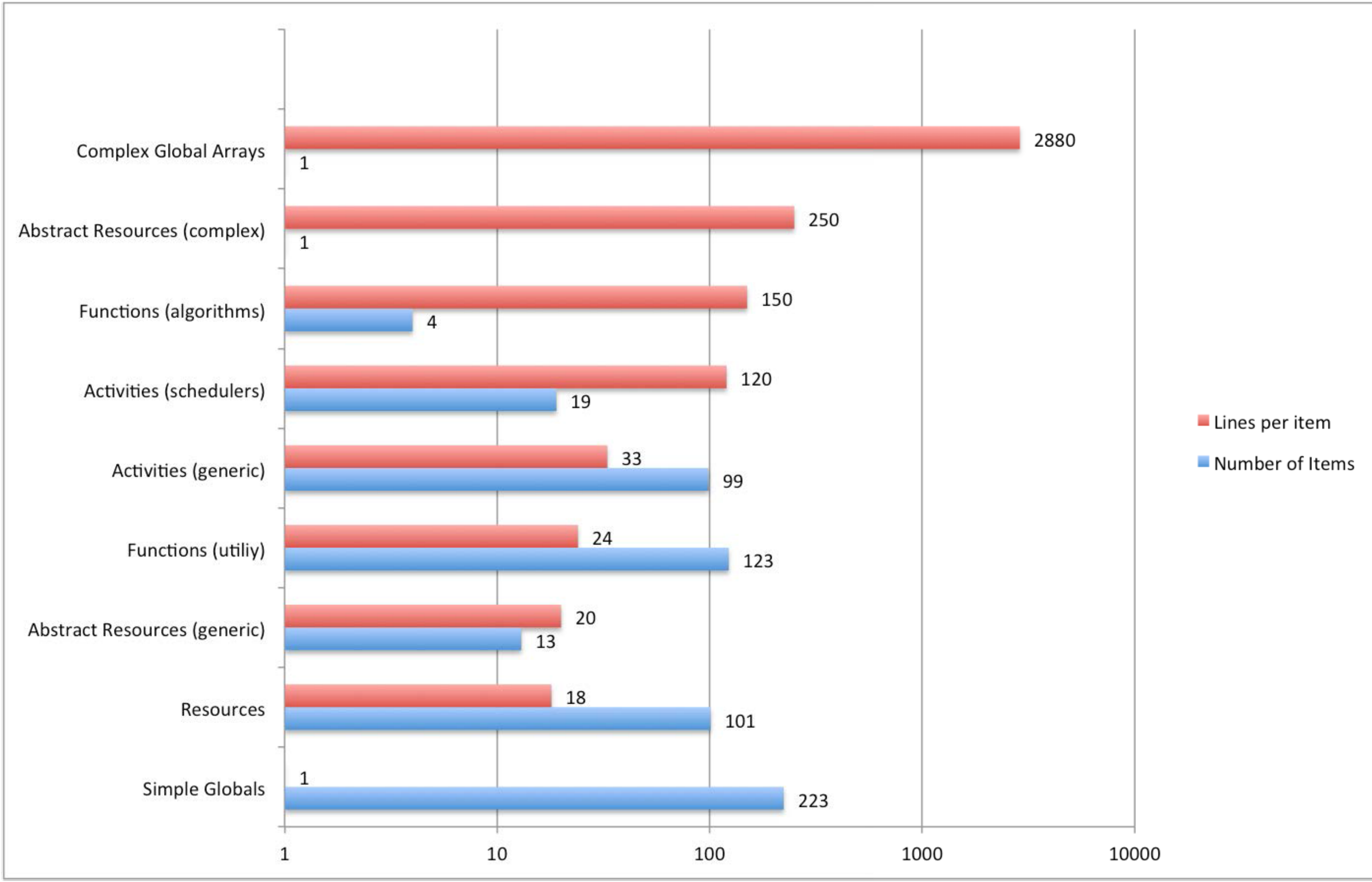
Patterns and automation in the Juno adaptation

- Much of the adaptation code was inherited from previous missions
- About half of the schedulers involve features that are specific to the Juno S/C
- Adaptations such as AUTOGEN would benefit from true adaptation templates
- Currently, re-using a previous adaptation of APGEN means a lot of hand-editing

MRO

Activities that are automatically scheduled by AUTOGEN
Orbital geometry events: Occultations, Eclipses, Periapsis and Apoapsis, Ascending and Descending Nodes
Daily activities
Reaction wheel assembly desaturation activities
HGA (Hi Gain Antenna) hard stop activities
HGA management activities
Ranging and radio science orbits
Downlink data rate selection
Low-elevation downlink suppressors
Communications blocks
Daily activities in contact
Critical low-level activities

MRO statistics



Patterns and improvements in the MRO adaptation

- Many of the schedulers are based on generic logic, although activity specifics (names, parameters) are mission-specific
- Good opportunity to introduce templates into the APGEN DSL
- The API for searching for windows of opportunity could be improved

MSL

- The MSL mission was not planning to make use of APGEN
- An opportunity to use APGEN showed up during the Cruise phase for two reasons:
  - scheduling telecommunications sequences was a very repetitive activity
  - such sequences were becoming more and more critical as the S/C approached EDL
- Shows the adaptability of the APGEN-based ground model: a prototype was generated in only a few days

Outline

- Introduction to APGEN
- Resource modeling in early versions of APGEN and SEQGEN
- Evolution of the APGEN adaptation language
- Application of APGEN scheduling to space missions
  - Generic morphology of an APGEN adaptation
  - DI/EPOXI
  - Europa (proposed mission)
  - INSIGHT
  - Juno
  - MRO
  - MSL
- Conclusion

Conclusion

- We have identified the key elements of a "generic APGEN adaptation"
- Our findings will pave the way for future improvements in three areas:
  - turning to distributed processing for easier integration with external models and other uplink components
  - improved DSL to flatten the learning curve
  - templates for facilitating re-use of existing adaptations

Acknowledgments

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

Backup Slides

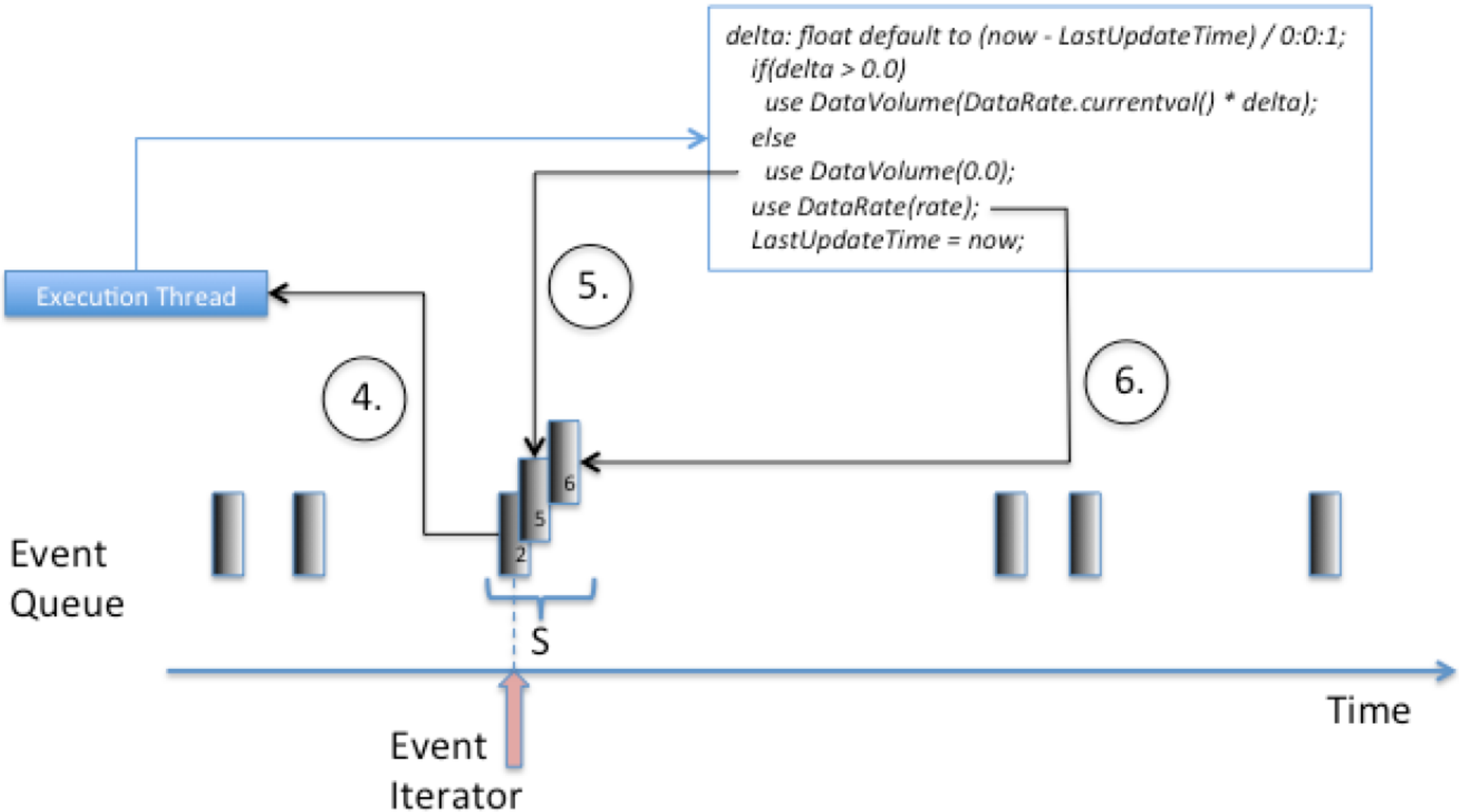
Abstract resource code, using *a priori* modeling

```
resource AddData: abstract
 begin
  parameters
   rate: float default to 0.0;
  resource usage
   use DataVolume(DataRate.currentval()
     * ((now  - LastUpdateTime.currentval()) / 0:0:1))
     when LastUpdateTime.currentval() < now;
   use DataVolume(0.0) when LastUpdateTime.currentval() >= now;
   use DataRate(rate);
   use LastUpdateTime(now);
  end resource AddData
```

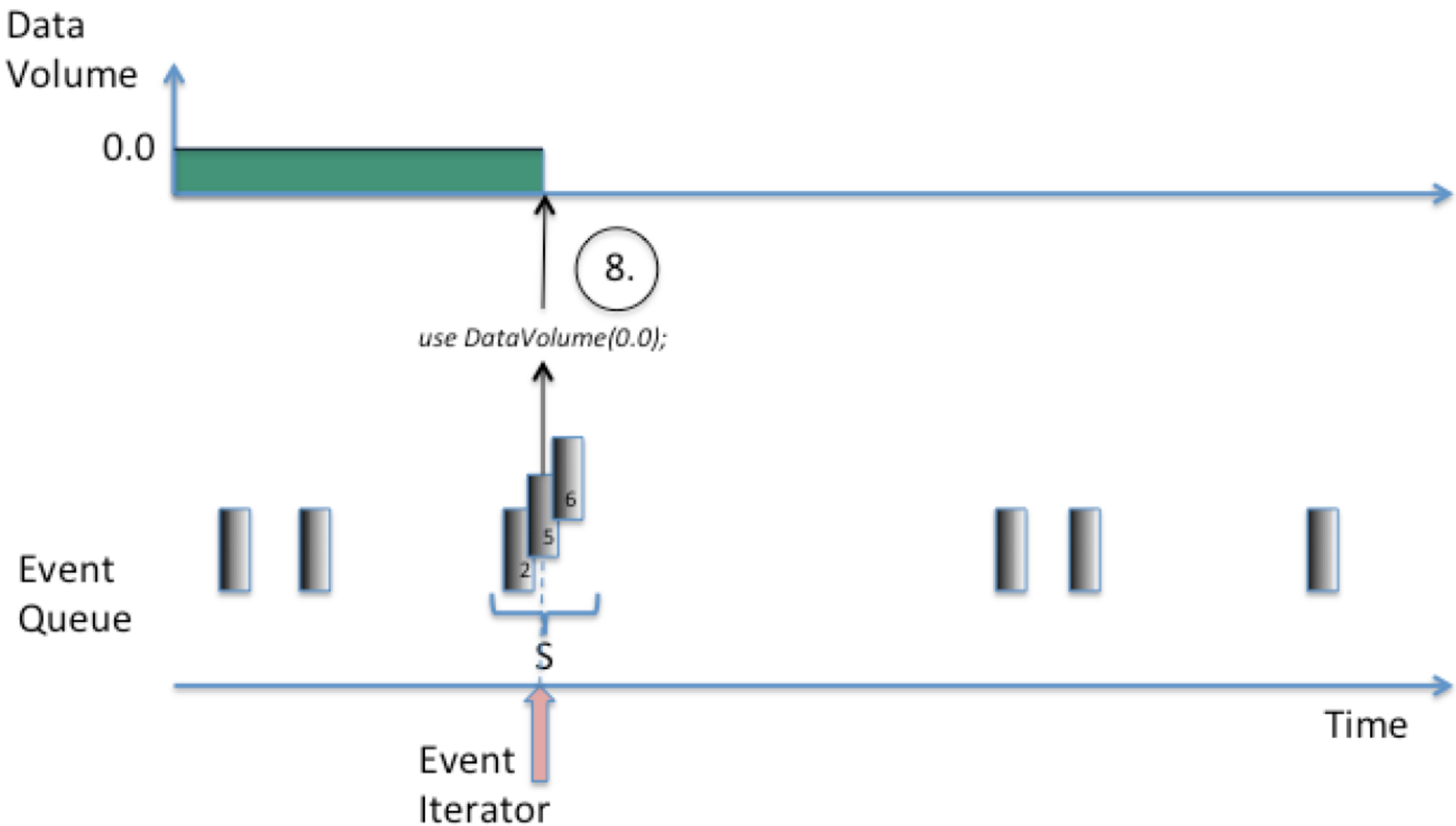
Abstract resource code, using concurrent modeling

```
global time LastUpdateTime = 2020-001T00:00:00;
resource AddData: abstract
begin
   parameters
      rate: float default to 0.0;
   modeling
      delta: float default to (now - LastUpdateTime) / 0:0:1;
      if(delta > 0.0)
         use DataVolume(DataRate.currentval() * delta);
      else
         use DataVolume(0.0);
      use DataRate(rate);
      LastUpdateTime = now;
end resource AddData
```

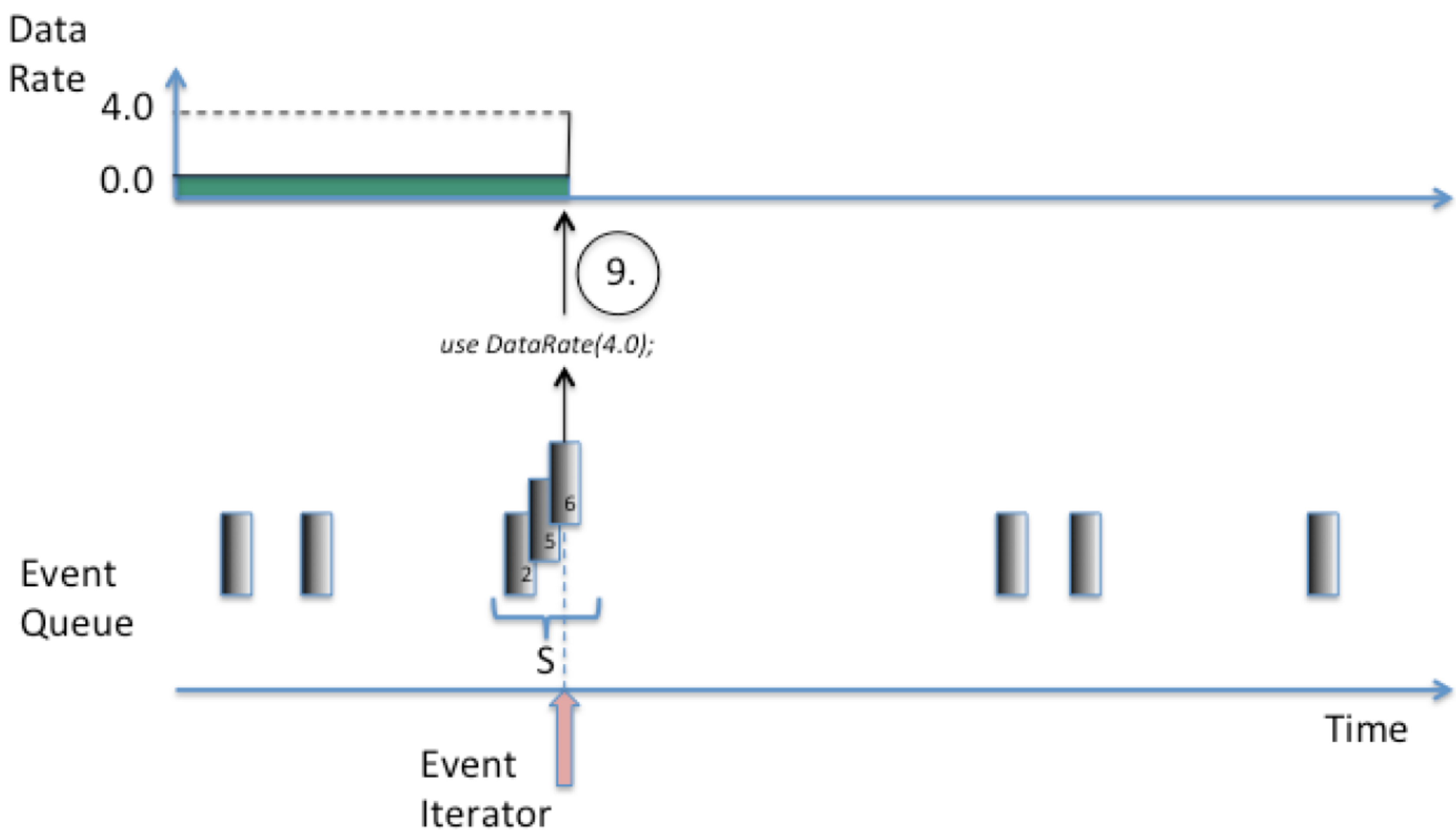
How the code gets executed (1)



How the code gets executed (2)



How the code gets executed (3)



Extending the APGEN DSL through external libraries

- JPL domain experts have developed a number of powerful simulation tools over the years, e. g.:
  - SPICE toolkit
  - MMPAT Multi-Mission Power Analysis Tool
  - Slewth, an ACS simulation tool for the DAWN mission
  - Telecom Forecast Predictor (TFP)
  - …
- APGEN and SEQGEN adapters need to leverage this modeling capability into their adaptations
- "Glueware" in the form of user-defined libraries has been provided to let adapters achieve their goal

Introduction of a scheduling capability

- New type of activity: scheduler
- Scheduling algorithm features two passes
  - first, "modeling" pass involves all normal activities and computes all S/C and ground states
  - in a second pass, schedulers are allowed to run also
  - schedulers have access to a method that computes windows during which a certain condition is satisfied
  - windows can extend into the future thanks to the availability of state values calculated in the first pass
- More details are available in the paper