

MCRunjob:

An HEP Workflow Planner for Production Processing

Greg Graham
CD/CMS Fermilab
CD Projects Meeting
3-Apr-2003

Outline

- Introduction to the Software
- Relationship to Experiments
- Scope of the Proposed Project
- Relationship to Other Projects

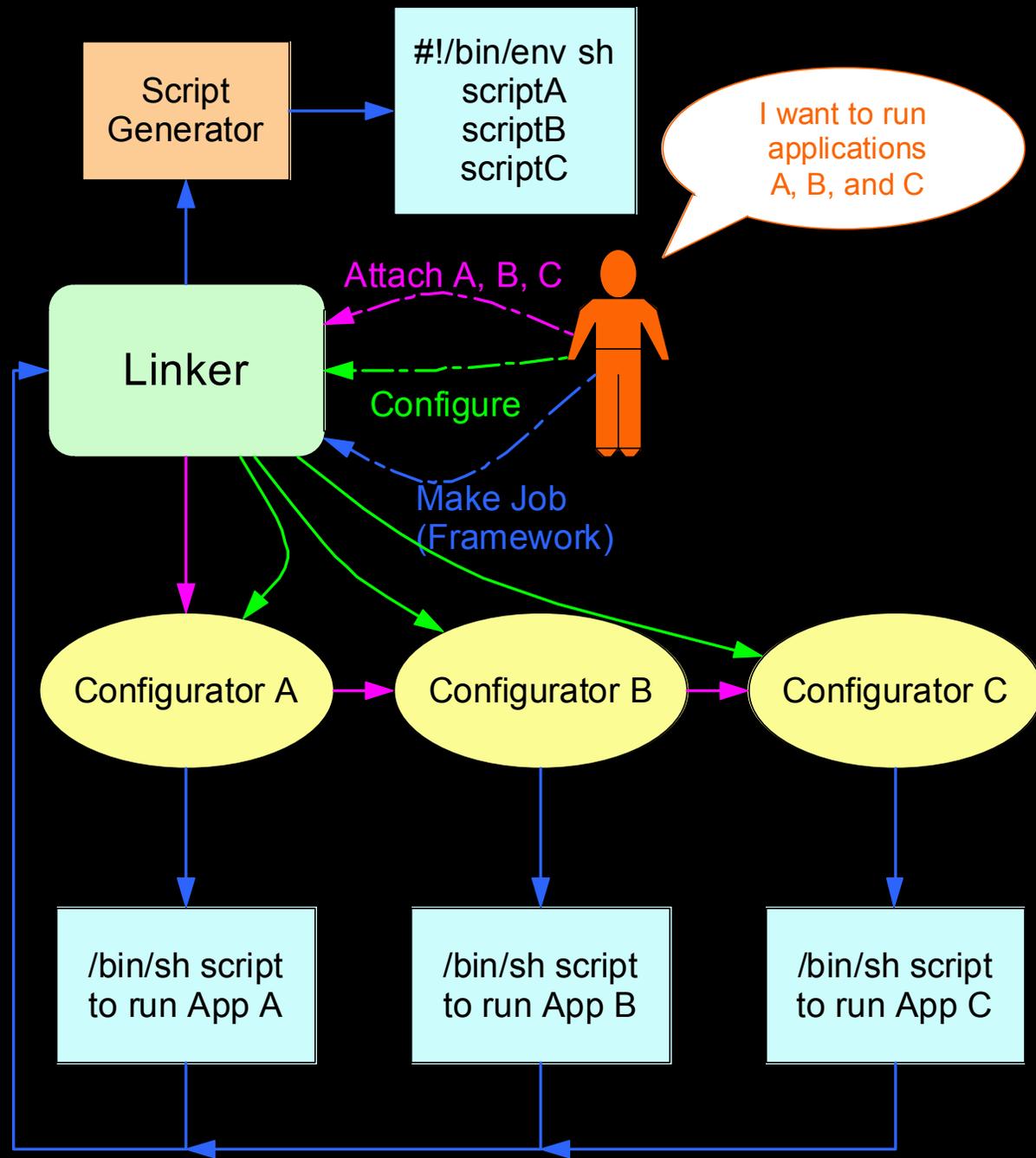
Purpose of MCRunjob

- Applications in complex production processing environments often need to be tamed
 - Hundreds of input parameters encountered during MC Production
 - Heterogeneous runtime environments, many different Regional Centers
 - Complex multi-application workflows, spanning both MC production and Analysis
 - Dependencies and relationships among the metadata *are often modeled inside of obscure shell scripts*
- MCRunjob captures specialized knowledge involved in workflow planning and makes it available to users and to higher level tools.
 - Metadata and schema oriented descriptions of workflow components
 - Tracks *dependencies* among the metadata
 - Tracks synonyms between groups of metadata, allows schema evolution and versioning
 - User registered functions do the actual work within a framework, leading to enhanced modularity
- MCRunjob has been in use since 1999 at DZero and since 2002 at CMS: This is a mainstream tool already supported by the respective experiments

A user who wants to run applications A,B, and C attaches corresponding Configurators to a Linker. The Linker verifies that dependencies are satisfied.

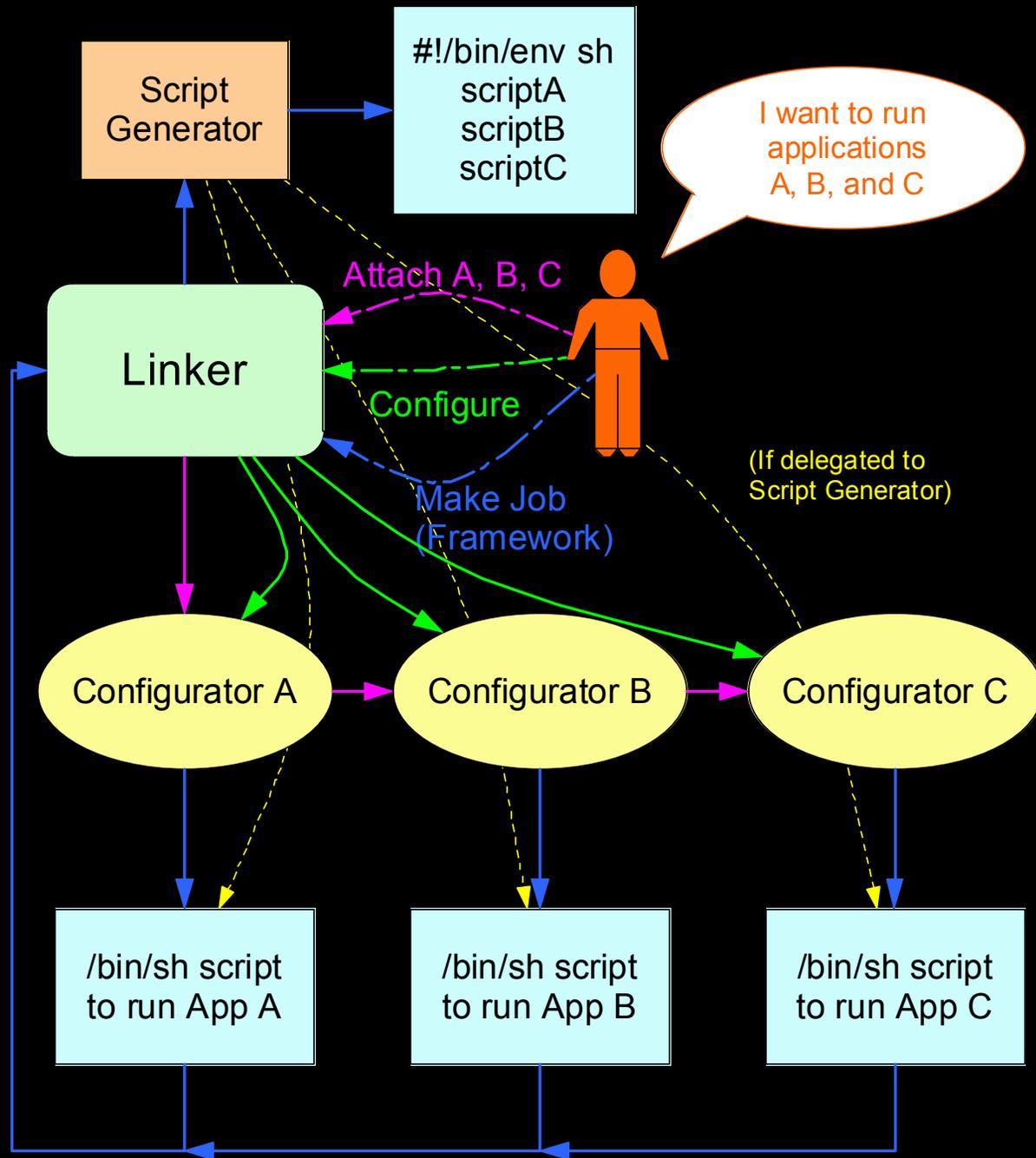
Once attached, the user sets values for the various schema elements defined in each configurator, and defines filename rules, random seed rules, etc.

The user then executes the framework. Each Configurator may generate scripts used to run the corresponding application. The scripts are collected by a ScriptGen object.



The ScriptGen object is obviously a very specialized component. Therefore, Configurators are able to delegate framework handlers to ScriptGen objects. This allows script generating code that targets specific environments to be collected in a single ScriptGen module.

Multiple ScriptGen objects can be attached at once, allowing two different environments to be targeted by the same workflow description.



Modularity

- Metadata is handled internally in a modular way
 - Configurators can hold related schema elements together that describe a task, application instance, external service, catalog, or database.
 - Configurators can access external services using the same schema and metadata oriented semantics already developed
- Script generating Configurators are also modular
 - Configurators that generate jobs are modular in that they can be delegated to handle all script generating functions, so that the “target” scripts can be selected by including the appropriate module.
 - Eg- ImpalaScriptGen, VDLScriptGen, MOPDagGen, etc.
- Configurator authoring is easy because much of the work is done by simple, easily defined functions that are registered into the Configurator.
 - Eg. Macro parsing, Framework Call handling, stored functions, etc.

Use Cases - Current

- Generation of complex multi-application workflows. Derived data products may require multiple processing steps. MCRunjob chains individual steps together into tree structures, and allows for logical and functional dependencies to be declared among the metadata and groups of metadata. (For example, “Use the Run Number for the Random Number Seed,” or “Filename=‘Metadata1_Metadadata2.Metadadata3’ ”)
- Planning of Monte Carlo requests. MCRunjob can be used to split a large request into multiple smaller requests tailored for a specific farm or situation.
- Tools to give non-experts access to full spectrum of applications and services. MCRunjob exposes all applications, tasks, and services to the user as metadata and takes care of hiding regional center or runtime environment eccentricities.

Use Cases - Current

- Bringing a new Regional Center On Line Quickly. As a part of the overall package of software needed to install an experiment software environment at a Regional Center, MCRunjob can be used to quickly test the installation by running the actual applications in complex workflows right away. MCRunjob can expose a uniform interface that hides Regional Center differences.
- Integration of Applications with experiment databases. MCRunjob can be used to generate metadata needed to track derived or created data products as part of the workflow description. MCRunjob can also be used to retrieve production processing requests from experiment databases. (In DZero, this is SAM; in CMS it is the RefDB.)

CMS and DZero Computing

- MCRunjob is the official tool for Monte Carlo Production at DZero and CMS.
 - Used to generate millions of events at DZero regional centers worldwide.
 - Used to generate 1.5M events on the CMS Integration Grid Testbed using MOP and Condor-G. (WBS 1.3.3)
- CMS and DZero continue to work on experiment specific extensions to MCRunjob.
 - SAM/JIM execution environment in DZero, runtime extensions
 - Inclusion of new runtime scripts (CMSProd) and different Grid environments in CMS (MOP, VDL, etc) (WBS 1.3.2, 1.3.3)
- Common code project started this year: Shahkar
 - holds common base classes for MCRunjob

MCRunjob Core Services

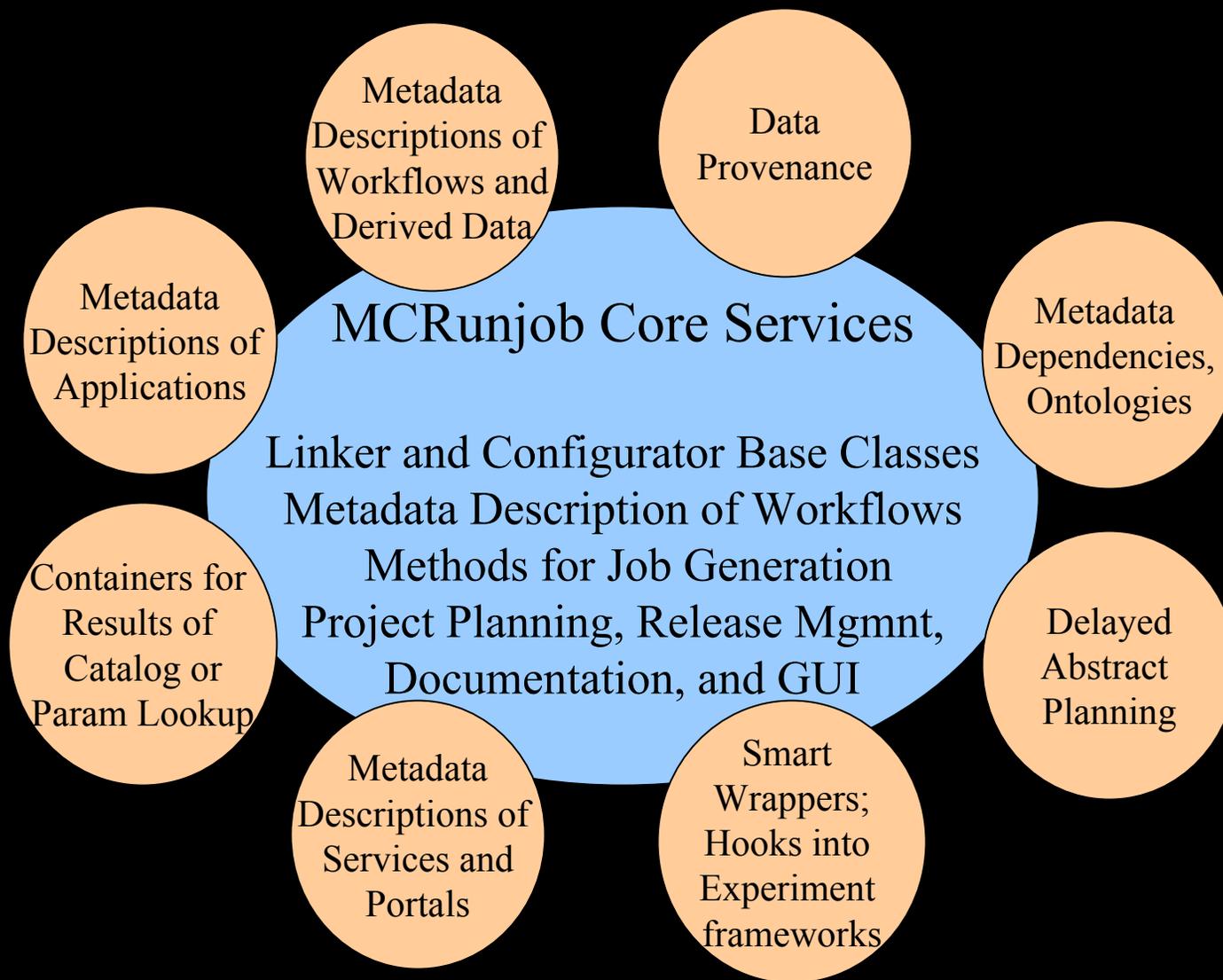
Linker and Configurator Base Classes

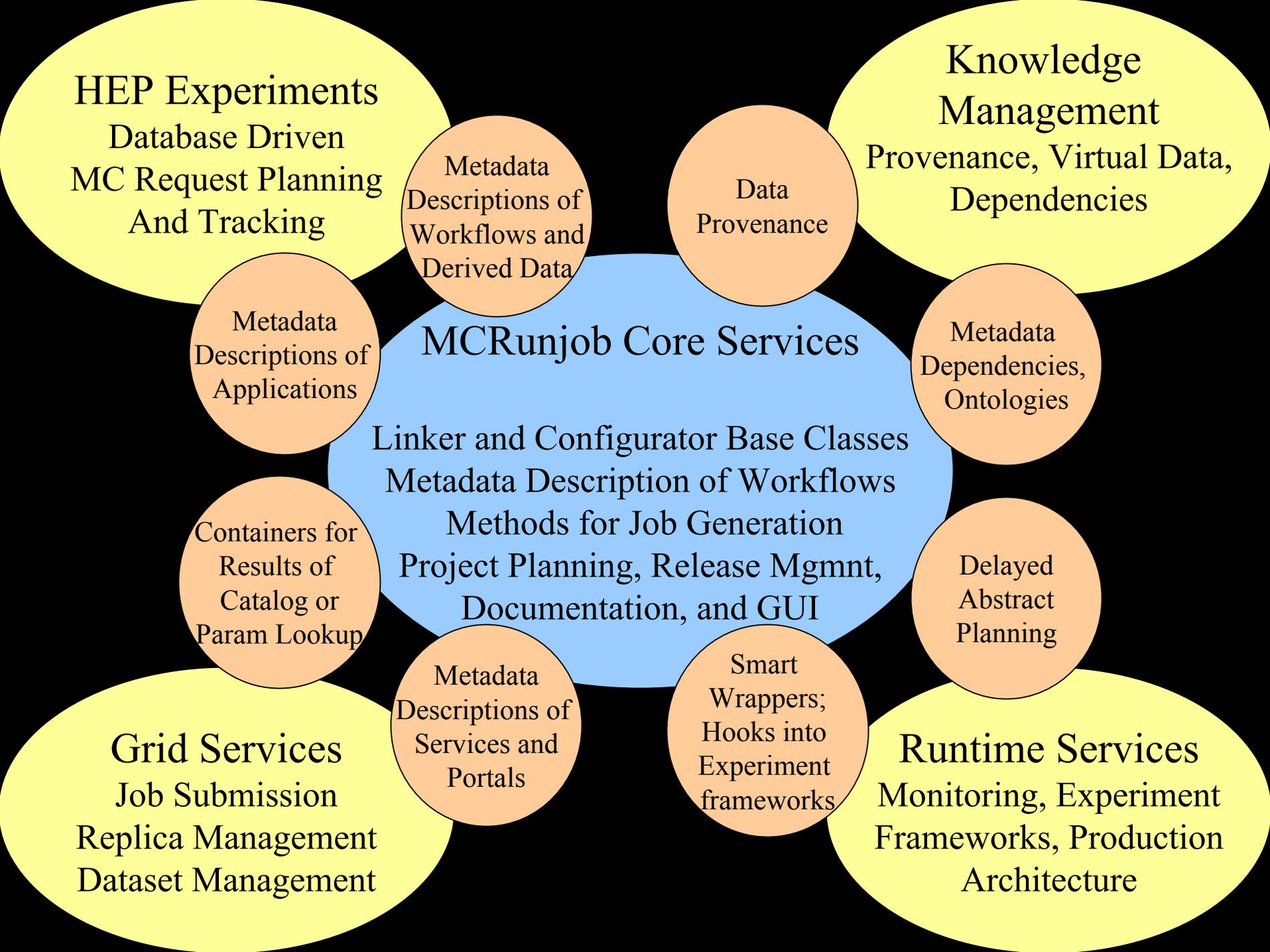
Metadata Description of Workflows

Methods for Job Generation

Project Planning, Release Mgmt,

Documentation, and GUI





HEP Experiments
Database Driven
MC Request Planning
And Tracking

Knowledge Management
Provenance, Virtual Data,
Dependencies

Metadata
Descriptions of
Workflows and
Derived Data

Data
Provenance

Metadata
Descriptions of
Applications

MCRunjob Core Services

Metadata
Dependencies,
Ontologies

Linker and Configurator Base Classes
Metadata Description of Workflows
Methods for Job Generation
Project Planning, Release Mgmt,
Documentation, and GUI

Containers for
Results of
Catalog or
Param Lookup

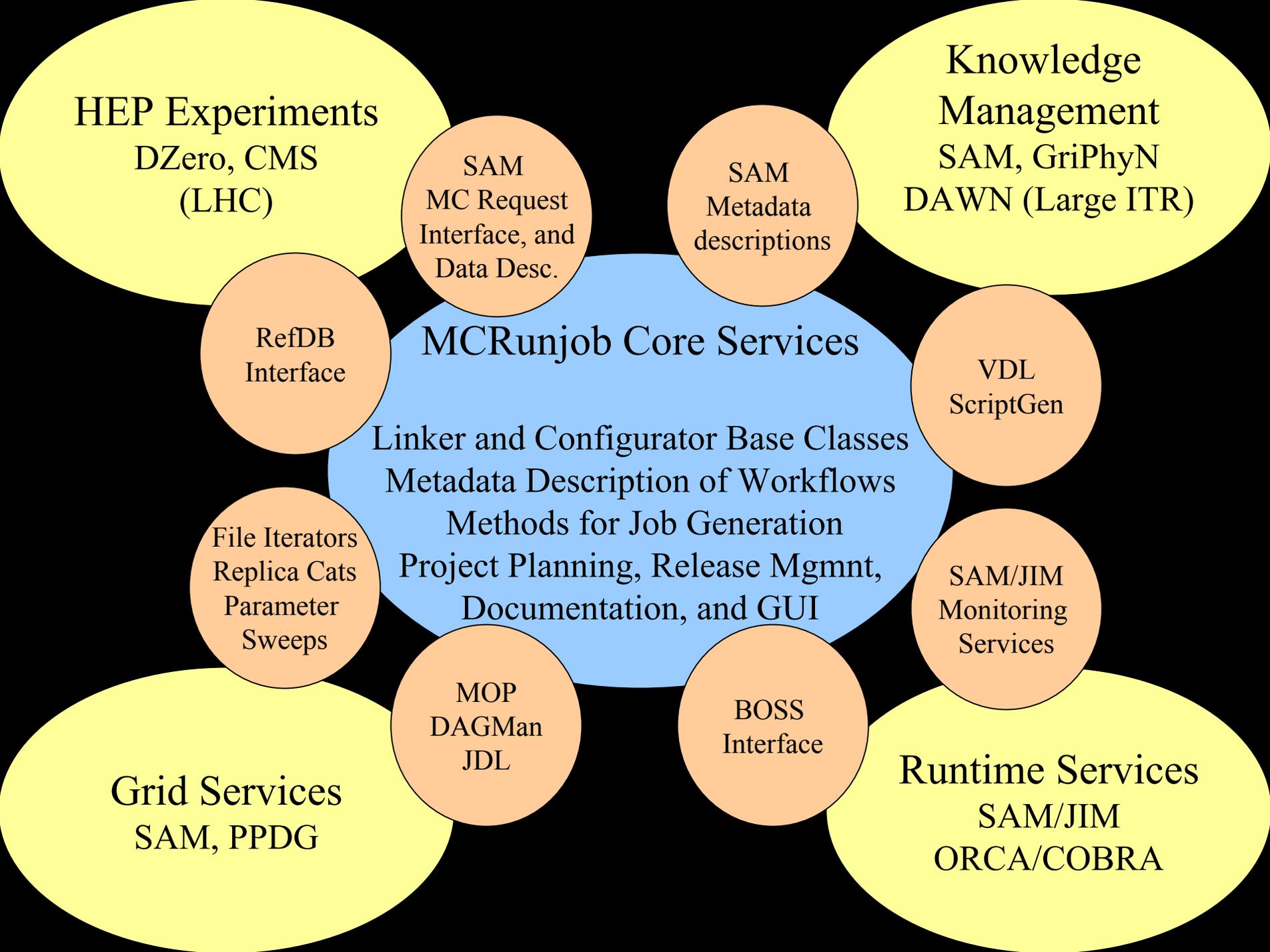
Delayed
Abstract
Planning

Grid Services
Job Submission
Replica Management
Dataset Management

Metadata
Descriptions of
Services and
Portals

Smart
Wrappers;
Hooks into
Experiment
frameworks

Runtime Services
Monitoring, Experiment
Frameworks, Production
Architecture



HEP Experiments
DZero, CMS
(LHC)

SAM
MC Request
Interface, and
Data Desc.

SAM
Metadata
descriptions

Knowledge
Management
SAM, GriPhyN
DAWN (Large ITR)

RefDB
Interface

MCRunjob Core Services

VDL
ScriptGen

Linker and Configurator Base Classes
Metadata Description of Workflows
Methods for Job Generation
Project Planning, Release Mgmt,
Documentation, and GUI

File Iterators
Replica Cats
Parameter
Sweeps

SAM/JIM
Monitoring
Services

Grid Services
SAM, PPDG

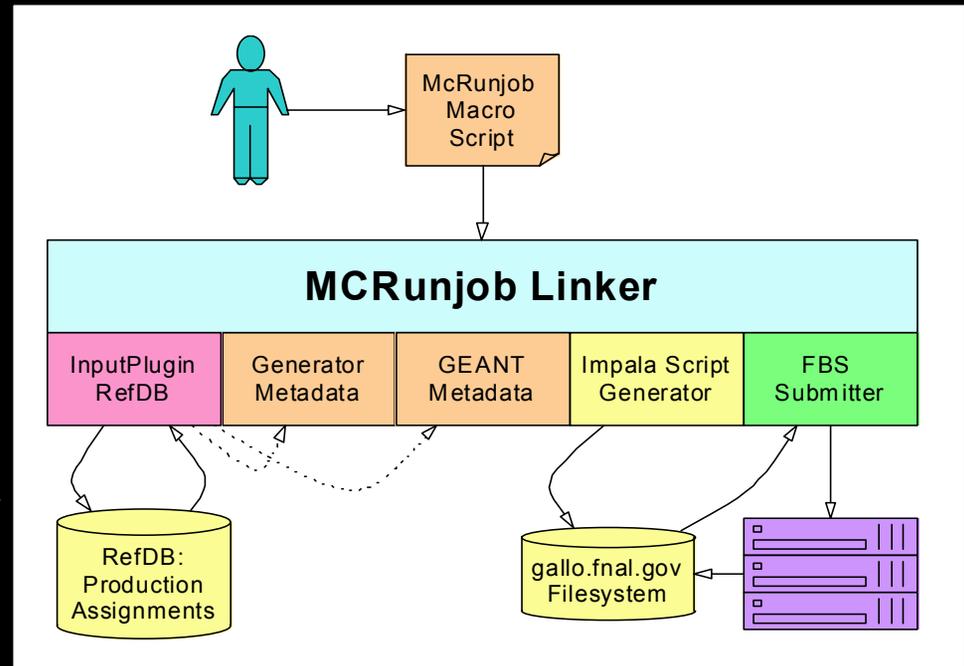
MOP
DAGMan
JDL

BOSS
Interface

Runtime Services
SAM/JIM
ORCA/COBRA

External Services and Configurators

- LNameStreamConfigurator, SAMStreamConfigurator
 - Can register a function to this Configurator that will fill a LogicalNameList with names (eg- LFNs, PFNs)
 - During framework operation, this Configurator will iterate over the list, setting the schema element “OutputSpec” to the current value.
- InputPluginConfigurator
 - InputPluginBashFile will parse environment variable definitions in a sh script and expose these by including the symbols as schema elements with the corresponding values
 - InputPluginRefDB will obtain schema elements and values from a web server with database backend



Other Configurators

- RogueConfigurator
 - No schema whatsoever- user defines it all at runtime!
- TableConfigurator
 - Derives from LNameStream, but has multiple schema elements. Can read from a table file or a database table and iterate over the rows
- ParamSweepConfigurator
 - Similar to a TableConfigurator, but has added logic to generate its own table internally according to some rules.
- VDLScriptGen
 - Produces Chimera Virtual Data Language output
- MOPDagGen
 - A ScriptGen Configurator that takes scripts generated by other ScriptGens, turns them into DAG nodes, and creates a master DAG.
- RunJobConfigurator
 - Takes specified script object, submits it to batch interface or grid portal.

Project Definition

- Scope of Project
 - Support of core functionality needed by the experiments DZero and CMS
 - Chaining individual production processing descriptions together to form complex workflow descriptions
 - Modularity to produce executable jobs from workflow descriptions for a variety of runtime environments, grid portals, and regional centers.
 - General APIs for connecting to experiment specific processing request DBs and tracking DBs
 - Common code project already started, code named “Shahkar”
 - Support efforts to extend functionality of MCRunjob to include more interfaces to Grid Services
 - Build upon current experience with MOP and DAGMan. Extend this methodology to modules for JDL and work closely with the LCG.
 - Extend the current experiences with File Iterators to make use of Replica Catalogues.
 - Extend the RunJob configurators to expose Grid Portals to MCRunjob

Project Definition

- Scope of Project (cont'd)
 - Support efforts to extend the work on Knowledge Management issues
 - Refine what is already there for Virtual Data Language by allowing MCRunjob generated scripts to be used as Chimera transformations.
 - Refine the MCRunjob macro language to make it more terse, support expressions, and more flavors of dependencies
 - Perhaps embedding the existing MCRunjob Macro Language directly in Python.
 - Support efforts to explore the possible runtime applications of MCRunjob
 - Runtime monitoring in SAM/JIM context
 - Write extension configurators to insert monitoring wrappers into existing workflows in a generic way.
 - Bridge experiment application frameworks to external services, such as Grid or experiment supported DBs.
 - Interested in talking to GANGA about their experiences.

Effort

- Current Effort on Core Project:
 - DZero maintenance and extensions: 1 FTE
 - Dave Evans, GridPP, Lancaster
 - CMS maintenance and extensions: 1 FTE
 - Julia Andreeva, Veronique Lefebure, and Nikolai Smirnov, CMS CCS and INFN
 - CMS Grid Extensions: 0.5 FTE
 - Anzar Afaq, PPDG
 - CMS and Shahkar release management, testing, planning : 0.25 FTE
 - Greg Graham, USCMS
- Common code repository is going VERY slowly
- As development continues where needed, documentation and software quality suffers.
 - SAM/JIM April milestones, MOP DAGMan interface on CMS, etc...

Effort

- Requesting 1 FTE from CD to work on MCRunjob Core Project
 - Coding Quality and Common Code Project
 - Release Management and Testing
 - Documentation
 - GUI
- Pulls in effort from other projects at a very exciting time
 - PPDG: MOP, Condor-G
 - GriPhyN: Chimera, Virtual Data Language
 - DAWN: Knowledge Management Tools
 - Runtime and Experiment Frameworks
- Synergy between CMS and DZero, who already use this tool
 - Code bases will continue to diverge if not arrested soon
 - Shahkar is started, but moving too slowly
- Challenges coming up quickly
 - CMS DC04 50M events, DZero SAM/JIM April milestone

Relationship to Other Projects

- SAM
 - One of the first great applications of MCRunjob was to automatically generate the metadata needed by the SAM system in order to store MC production results.
 - Closer integration with SAM is proceeding apace in the context of automatic generation of MC jobs from request metadata stored in SAM
- CHIMERA
 - MCRunjob has a ScriptGen which produces Virtual Data Language
 - Conceptually, Configurator schemas are like transformations, Configurators with values are like derivations, and ConfiguratorDescriptions and dependencies define “types” on the data appearing at the endpoints of a transformation.
 - MCRunjob can either generate VDL, VDL+wrapper scripts (custom transformations), or function as an abstract planner.

Relationship to Other Projects

- SAM/JIM
 - In the JIM grid execution environment, “abstract” MCRunjob scripts are sent as the job instead of shell scripts or conventional executables.
 - MCRunjob macro scripts are re-parallelized by a remote MCRunjob Linker process started up by Condor-G.
 - Delayed abstract planning!
- Data Provenance
 - MCRunjob *is already capable of a fully declarative specification of workflow*, and can communicate with external databases and servers.
 - Besides a bare specification of parameters, MCRunjob keeps track of the dependencies that existed among parameters when they were created.
 - Expected to play a role in DAWN large ITR

Conclusions/Questions

- MCRunjob provides functionality to model complex workflows found in MC Production.
- MCRunjob is a powerful workflow planner with modular component based interfaces to external services.
- Metadata from any one application area should be exposed to the other areas without compromising existing architectures
- Should be able to run jobs on any resources in a unified way
- In preparation for Analysis environments:
 - Take it from a former Kaon physicist: Sharpening our understanding of coarse grained production processing still has much to teach us about the more complex environments expected in physics analysis.
 - Understanding the behavior of the underlying Grid services and the coming challenges of knowledge management in the face of clean predictable input and measurable results has a lot of value.

References

- USCMS MCRunjob page :
 - <http://www.uscms.org/scpages/subsystems/DPE/Projects/MCRunjob/>
- DZero MCRunjob page :
 - <http://www-clued0.fnal.gov/runjob/>
- Previous Talks and Papers:
 - MCRunjob: A Workflow Planner for HEP, G.E. Graham, Dave Evans, and Iain Bertram. Proceedings of Computers in High Energy Physics 2003 (CHEP 2003), San Diego, CA
 - Tools and Infrastructure for CMS Distributed Production (4-033), G.E. Graham, et al. Proceedings of Computers in High Energy Physics 2001 (CHEP 2001), Beijing, China
 - Dzero Monte Carlo Production Tools (8-027), G.E. Graham, et al.. Proceedings of Computers in High Energy Physics 2001 (CHEP 2001), Beijing, China
 - Dzero Monte Carlo, G.E. Graham. Proceeding of Advanced Computing and Analysis Techniques 2000 (ACAT 2000), Fermilab, Batavia, IL
- ggraham@fnal.gov evansde@fnal.gov