



KINETIC REQUEST

Pumping Up With KURL

The Kinetic Uniform Request Language (KURL) is a domain-specific language written in Ruby that allows service providers and internal service functions to automate the creation and development of service items defined and categorized in spreadsheets and other formats. KURL automation allows service providers to build out services for one company or constituency, extract them into KURL, and easily reuse them within different environments. The result is faster time-to-value and considerably reduced costs.

IT outsourcers and other types of service providers create value for their customers by delivering thousands of different service items more efficiently and for less than it would cost customers to do these things for themselves. Many of these services are similar from company to company and their requirements are often well defined in spreadsheets and other formats.

The market for outsourced services is highly fragmented and rapidly growing. In the IT space alone, there are thousands of service providers, ranging from Gartner's Top 20, with revenues of \$9 billion to \$60 billion, to hundreds of such businesses in the INC 5,000, with revenues in the \$2 million to \$100 million range.

One key differentiator in this competitive marketplace is the ability to quickly deliver higher value to customers by offering specific service items more quickly and cost-effectively than competing service providers.

There is no better way to do this than by automating the creation and deployment of service items and service catalogs. Such a strategy involves developing reusable and transportable service items (prepackaged forms and processes) that can be deployed in any customer environment in a fraction of the time it takes to set up each customer's system individually. The benefits of automation include:

- Faster customer on-boarding for shorter time-to-value and higher customer satisfaction. Automation through reusable service items can help get customers up and running in one to two months versus six months or more.
- Higher profitability. Reusable and transportable service items can be customized and rolled out to multiple customers for a fraction of the cost of setting up customers individually.
- Verticalization. Adopting an automated strategy makes it easy to deliver service items to industry verticals without requiring in-depth IT expertise.
- For internal service organizations, such a strategy delivers higher demonstrable value to the enterprise by making service requests more easily and efficiently fulfilled.

Rapid Service Item Deployment in an Enterprise Request Management (ERM) Environment

Service request management has traditionally been “siloeed” within functional business areas or units, which often use different tools and processes to log, track, manage, and fulfill service requests.

Many external service providers typically take a similar approach by specialization—their systems and services are designed for specific functions, such as IT or accounting, and are seldom transferrable across functions.

Organizing service delivery in functional silos may seem logical from a business perspective, but it leads to several types of inefficiencies at the enterprise level. End users may be required to use different systems, or at least different user interfaces, to request services from different areas. This wastes time and increases training costs.

If the back-end systems required for service fulfillment don't "talk" to each other, error-prone manual data entry is required, leading to further inefficiencies and redundant, but potentially mismatched data in different systems, as well as potential scheduling conflicts.

Furthermore, in this service environment, enterprises are faced with complex service delivery models. In many cases, departments that are responsible for service delivery are required to leverage data and processes managed by systems "owned" by other functional areas.

Thanks to recent advancements in hardware and software interoperability, Enterprise Request Management (ERM) has today become a reality. ERM uses a common set of tools capable of accessing federated data and communicating with disparate departmental software applications to automate service request management across all units in an enterprise, or across any number of external customers. ERM typically consists of a configurable portal that provides a consistent user experience regardless of the type of service request, coupled with a task automation "backbone" that manages scheduling and approvals with various enterprise applications, as well as reporting tools for process management and optimization. The benefits of ERM include:

- Faster, less costly request processes due to a consistent user interface that requires no training;
- Elimination of manual data entry and duplicated data;
- Accelerated service delivery times;
- Ensuring resource availability and avoiding scheduling conflicts, so services are delivered correctly the first time;
- No more "rogue processes"—approvals are automatically requested and acknowledged before service requests can progress.

KINETIC DATA

Your business. Your process.



A = Approval, S = Schedule, F = Fulfillment

Enterprise Request Management typically consists of a configurable cloud-based portal that provides a consistent user experience regardless of the type of service request, coupled with a task automation “backbone” that manages scheduling and approvals.

Enable and automate ERM on a vast scale with KURL

Kinetic Request coupled with Kinetic Task is an ERM portal and workflow task automation system used by numerous large service providers and Fortune 500 companies. Service requests submitted through a Kinetic Request portal can serve as triggers to initiate ERM processes. Kinetic Task offers complete workflow control and flexibility with the ability to easily configure a limitless number of tasks and approvals to manage processes ranging from simple to the most complex.

One key feature of both Kinetic Request and Kinetic Task is their architecture, which allows users to create and configure service items and the workflows behind them by configuring tasks without programming. Instead, users deploy an intuitive GUI (called the Task Builder), which features task trees that enable automated workflows to be created with drag-and-drop simplicity.



Kinetic Task uses configurable decision trees to automate workflows without programming.

The Kinetic Uniform Request Language (KURL) is a domain-specific language developed by Kinetic Data to provide an alternative approach to creating service items in Kinetic Request and Kinetic Task. It is especially useful when the creation of service items is driven by volume. With KURL, service providers can rapidly and easily convert

service items in existing systems or in Excel and other formats into online service catalogs. And KURL files are portable, transferable and shareable from company to company and catalog to catalog, enabling service providers to easily create large volumes of service items with automated workflows for any number of customers.

How KURL works

KURL is a domain-specific language for Kinetic Request and Kinetic Task written in Ruby, a popular dynamic language with a focus on simplicity and productivity. A domain-specific language, unlike a general-purpose language, is designed to be useful for a specific task in a specific domain. In the case of KURL, the domain is Kinetic Request and Kinetic Task.

According to Microsoft, “By using Domain-Specific Language Tools, you can build customized modeling tools. You can define a modeling language and implement it very simply. For example, you can create a specialized language that describes a user interface, a business process, a database, or the flow of information, and then you can generate code from those descriptions.”^[1]

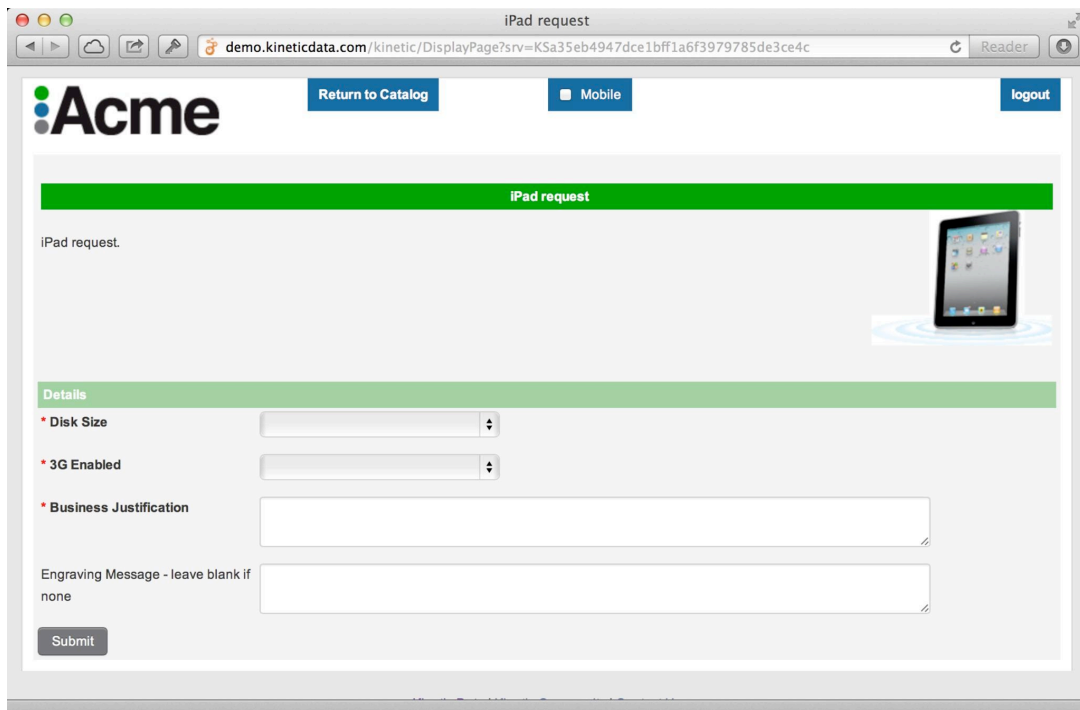
Wikipedia offers this analogy: “[Some languages] are like a knife. They can be used in thousands of different ways, from cutting food to cutting down trees. A domain-specific language is like an electric drill: it is a powerful tool with a wide variety of uses, but a specific context, namely, putting holes in things. A General Purpose Language is a complete workbench, with a variety of tools intended for performing a variety of tasks. Domain-specific languages should be used by programmers who, looking at their current workbench, realize they need a better drill, and find that a particular domain-specific language provides exactly that.”

KURL provides Kinetic Request and Kinetic Task users such a “drill” when the challenge is converting thousands of service items defined in spreadsheets and other formats into service items and service catalogs. KURL works by “parsing” spreadsheet data or HTML code into KURL text files and then by executing that parsed data into KURL codes to automatically generate new service items. In this manner, KURL can automatically generate thousands of service items defined in spreadsheets programmatically, as opposed to configuring services items individually in the Kinetic Task Builder.

Sample KURL code

```
service 'iPad Request'  
catalog 'ACME'  
image 'overview-wifi.png'  
description 'iPad request.'  
section 'Details'  
  
question 'Disk Size', :required, :choices=>["16Gb","32GB"]  
  
question '3G Enabled', :required, :choices=>["Yes","No"]  
  
question 'Business Justification', :required, :type=>"text"  
  
question 'Engraving Message - leave blank if none',  
  :type=>"text"  
  
standard_confirmation_page
```

KURL code turns predefined service definitions in spreadsheets and other formats into service items that are automatically fulfilled by Kinetic Task.



The screenshot shows a web browser window with the URL `demo.kineticdata.com/kinetic/DisplayPage?srv=KSa35eb4947dce1bff1a6f3979785de3ce4c`. The page is titled "iPad request" and features the Acme logo, "Return to Catalog", "Mobile", and "logout" buttons. The main content area has a green header "iPad request" and a sub-header "iPad request." with an image of an iPad. Below this is a "Details" section with four form fields: "Disk Size" (dropdown), "3G Enabled" (dropdown), "Business Justification" (text area), and "Engraving Message - leave blank if none" (text area). A "Submit" button is at the bottom left.

In this example, KURL code turned a service request description contained in a spreadsheet into a simple service request for an iPad.

For service providers, both internal and external, KURL means a large volume of service items can be created quickly and inexpensively. The three most common use cases for KURL include:

Catalog Provisioning. Using a catalog template and a suite of predefined services items to choose from, KURL can automatically gather requirements for a new catalog, and through its task automation, automatically build the new catalog.

Spreadsheet Parsing. If your service item requirements and standards are defined in spreadsheets, KURL can parse the spreadsheets and generate custom service items that can be cloned and re-purposed for any number of external or internal customers.

Service Item Request. In a manner similar to the use cases described above, KURL can automatically use existing service items in spreadsheets and other formats to gather necessary requirements, such as names and simple questions, to build additional service items through a KURL Task handler.

KURL in Action

By using KURL along with Kinetic Request and Kinetic Task, service organizations can provision any number of portals and create thousands of services items quickly and inexpensively. For example, one energy management and industrial controls company with over 100,000 employees worldwide used the spreadsheet-parsing capabilities of KURL to create over 2,000 service items in a fraction of the time and effort it would have taken to configure those service items individually. In another example, a large IT service provider won a multi-billion-dollar government contract by demonstrating the ability of Kinetic Request and KURL to provision and populate service catalogs for multiple federal agencies. Other service providers are using KURL to roll out managed services to different industry verticals using the same set of predefined services and requirements without investing heavily in industry domain knowledge.

Conclusion

Many large internal and external service providers need tools to automate the creation and deployment of service items and service catalogs, especially if they serve a large number of customers or face a high volume of service item requests. Using KURL with Kinetic Request may be the ideal solution for many companies, since it allows them to:

- Build out services for one company, extract them into KURL, and easily re-use them within a different customer environment;
- Deliver value faster at a lower price;
- Increase profitability by reducing costs of customer setup and ongoing service delivery;
- Roll out preconfigured common services like corporate “Genius Bars”; and
- Roll out services to vertical markets.

About Kinetic Data

Kinetic Data has helped hundreds of Fortune 500 and government customers—including General Mills, Avon, Intel, 3M, and the U.S. Department of Transportation—implement Enterprise Request Management (ERM), a holistic approach for centralized business process automation and service delivery management (SDM). Kinetic Data was named “Innovator of the Year” by enterprise software users, and it also has been recognized with awards for its superior customer service and support. Kinetic Data serves customers from its headquarters in St. Paul, Minn., offices in Sydney, Australia, and through a network of reseller partners.

[1] [http://msdn.microsoft.com/en-us/library/bb126235\(v=vs.80\).aspx](http://msdn.microsoft.com/en-us/library/bb126235(v=vs.80).aspx)