CUSTOM NOTIFICATIONS USING SEMANTIC MEDIAWIKI AND SERVER-SIDE SCRIPTING

Lexi McGillivray

NASA GRC-ATF

04-06-22

PRESENTATION OUTLINE

1. Overview of notifications at NASA GRC-ATF
   - The motivation
   - Establishing user-generated notifications
   - The evolution of need
2. Design
   - The new approach
   - Wiki implementation
   - Server-side implementation
3. Looking Forward
4. Q&A

Overview of notifications at NASA GRC-ATF

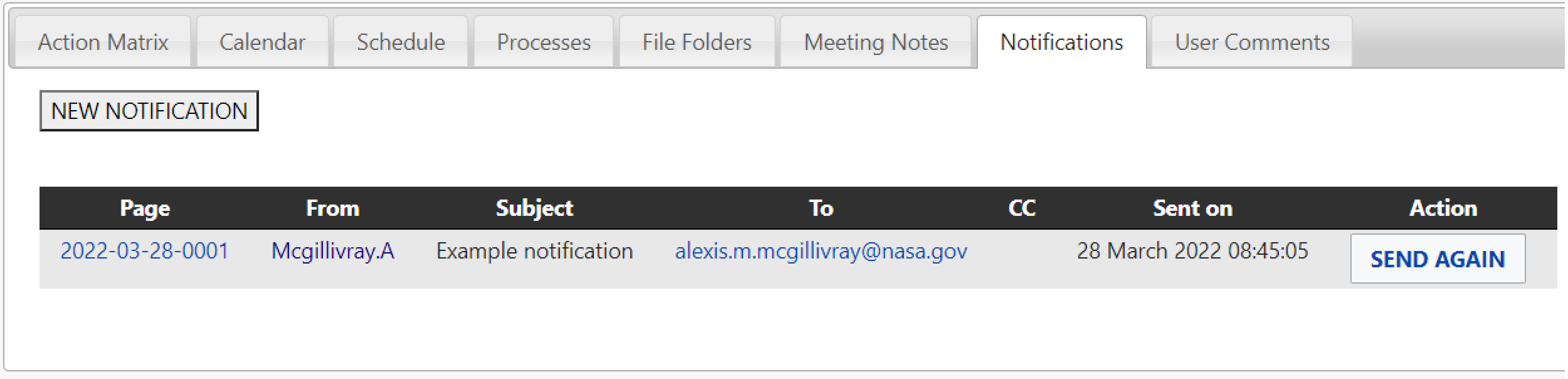
The motivation

The initial need was for user-generated notifications:

1. Our internal processes frequently require steps to be completed in sequence by more than one user, and communication was external to our wikis
2. Existing tools such as Mention, Category Watch, and Watchlist did not integrate well into our user base and processes

Overview of notifications at NASA GRC-ATF

Establishing user-generated notifications

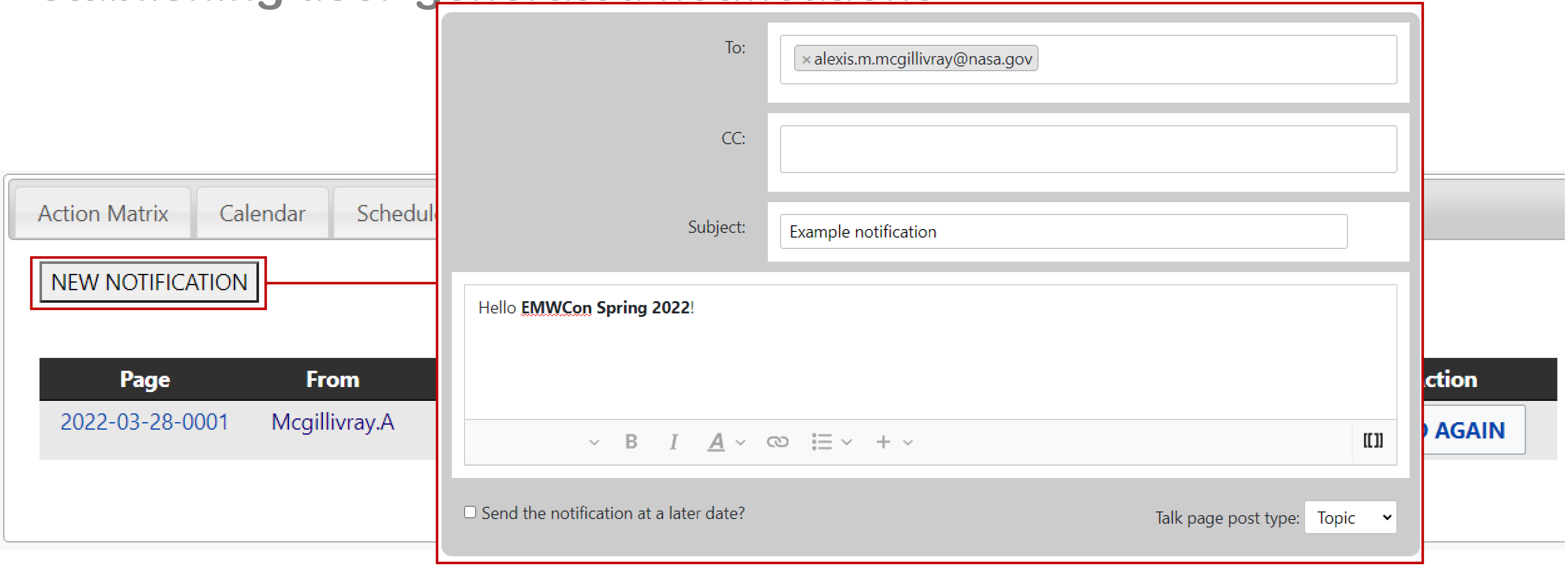


Page	From	Subject	To	CC	Sent on	Action
2022-03-28-0001	Mcgillivray.A	Example notification	alexis.m.mcgillivray@nasa.gov		28 March 2022 08:45:05	SEND AGAIN

We used Page Forms to define notification class pages, and incorporated them into our standard toolsets

Overview of notifications at NASA GRC-ATF

Establishing user-generated notifications



Page	From	ction
2022-03-28-0001	Mcgillivray.A	AGAIN

To: × alexis.m.mcgillivray@nasa.gov

CC:

Subject: Example notification

Hello **EMWCon Spring 2022**!

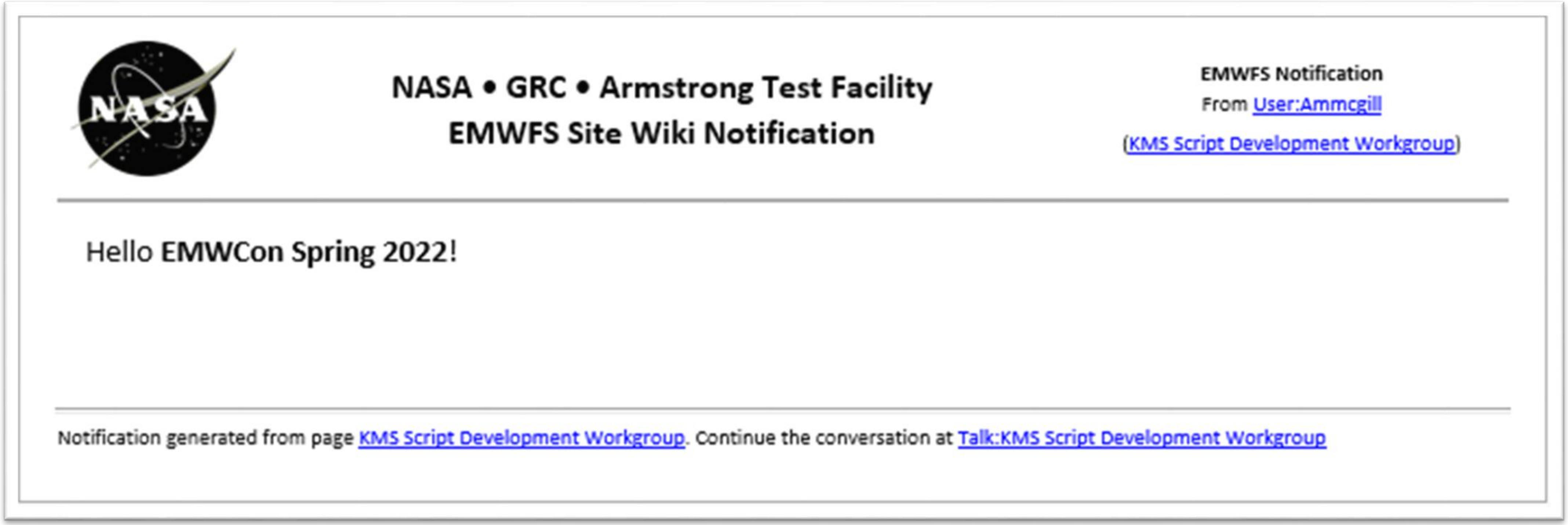
□ Send the notification at a later date?

Talk page post type: Topic

Buttonized input allows us to pre-define the recipients, subject, and body based on page class and properties

Overview of notifications at NASA GRC-ATF

Establishing user-generated notifications



NASA • GRC • Armstrong Test Facility
EMWFS Site Wiki Notification

EMWFS Notification
From User:Ammcgill
(KMS Script Development Workgroup)

Hello **EMWCon Spring 2022**!

Notification generated from page KMS Script Development Workgroup. Continue the conversation at Talk:KMS Script Development Workgroup

The email is generated and delivered by a script executed through a cron job running every three minutes. The message contains links back to the page from which it was generated, its associated talk page, and the user page of the sender

Overview of notifications at NASA GRC-ATF

The need evolves

Our user base found the notification tool to be very helpful in navigating to specific pages that were relevant to the work they were doing. Soon, they wanted event-based notifications to drive their work as well as those generated by users.

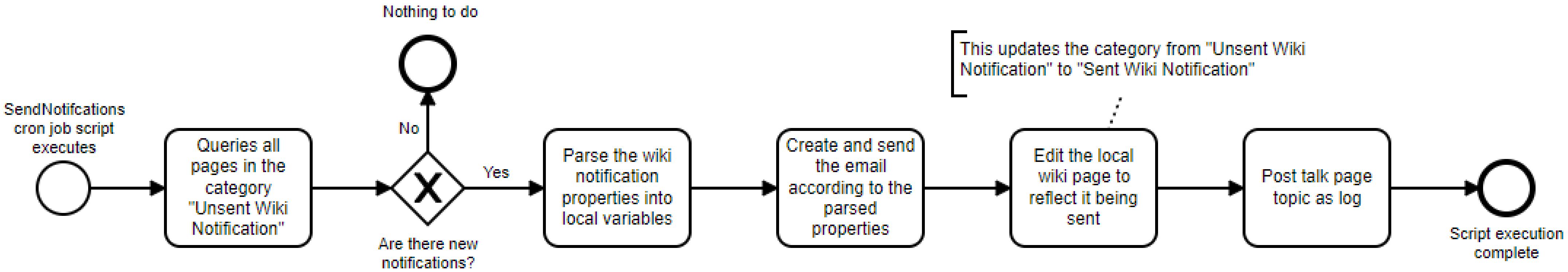
🖂 Reminders to review documents on a cycle

🖂 Site-wide notifications whenever a facility light status changes

🖂 Notification when an action has been completed and needs formal review

Overview of notifications at NASA GRC-ATF

The approach must evolve



SendNotifcations cron job script executes
Queries all pages in the category "Unsent Wiki Notification"
Nothing to do
No
Yes
Are there new notifications?
Parse the wiki notification properties into local variables
Create and send the email according to the parsed properties
This updates the category from "Unsent Wiki Notification" to "Sent Wiki Notification"
Edit the local wiki page to reflect it being sent
Post talk page topic as log
Script execution complete

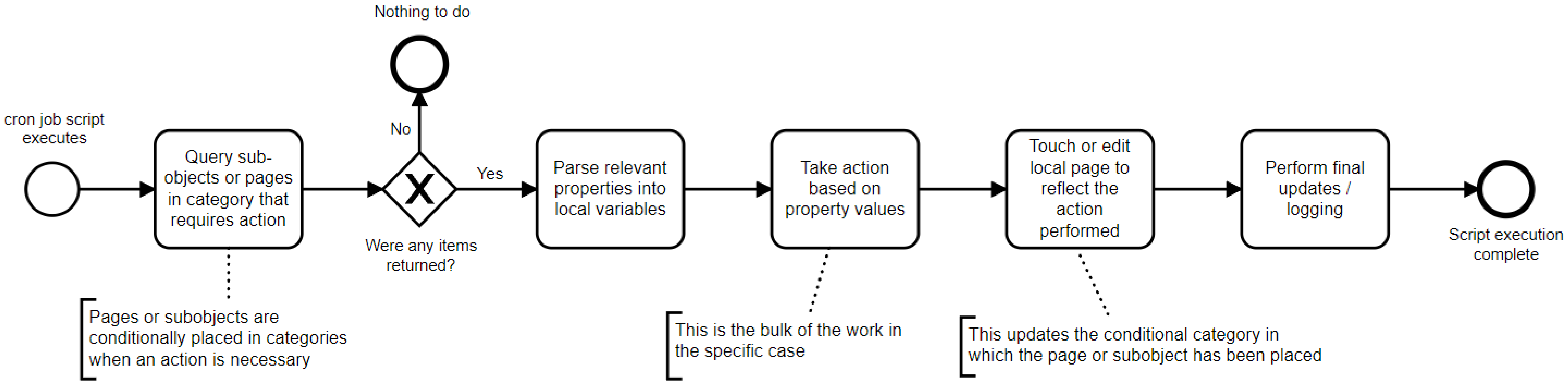
Simplified process for sending notifications

The shortcoming of this approach for the general case is how directly the function of creating notifications is tied to the Wiki Notification class and its instances, namely:

1. We don’t know how many automatic notifications a page will generate, so use of page properties becomes problematic
2. We don’t want to directly edit pages in different classes

Design

The New Approach

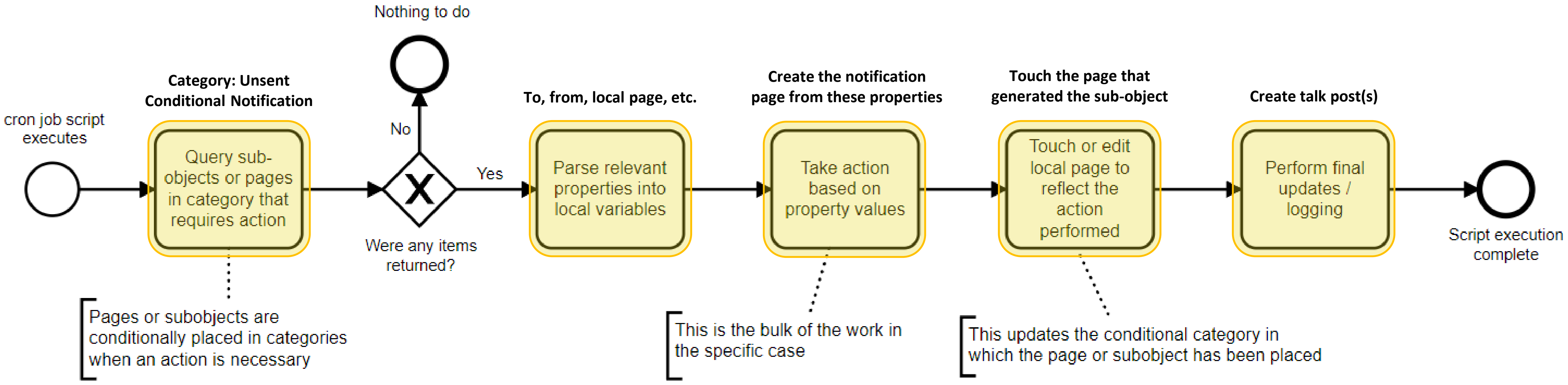

Nothing to do
cron job script executes
Query sub-objects or pages in category that requires action
No
Yes
Were any items returned?
Parse relevant properties into local variables
Take action based on property values
Touch or edit local page to reflect the action performed
Perform final updates / logging
Script execution complete
Pages or subobjects are conditionally placed in categories when an action is necessary
This is the bulk of the work in the specific case
This updates the conditional category in which the page or subobject has been placed

The general approach for triggering server-side scripts from a wiki page

1. Conditionally categorize pages using templates and sub-objects which can be re-categorized through a null-edit of a local page, or an edit of a dedicated page
2. Employ the MECE principle – we already have an approach to sending notifications, so we only need to generate them in a new way

Design

The New Approach



Nothing to do
cron job script executes
Category: Unsent Conditional Notification
Query sub-objects or pages in category that requires action
No
Yes
Were any items returned?
To, from, local page, etc.
Parse relevant properties into local variables
Create the notification page from these properties
Take action based on property values
Touch the page that generated the sub-object
Touch or edit local page to reflect the action performed
Create talk post(s)
Perform final updates / logging
Script execution complete
Pages or subobjects are conditionally placed in categories when an action is necessary
This is the bulk of the work in the specific case
This updates the conditional category in which the page or subobject has been placed

The general approach for triggering server-side scripts from a wiki page

1. Conditionally categorize pages using templates and sub-objects which can be re-categorized through a null-edit of a local page, or an edit of a dedicated page
2. Employ the MECE principle – we already have an approach to sending notifications, so we only need to generate them in a new way

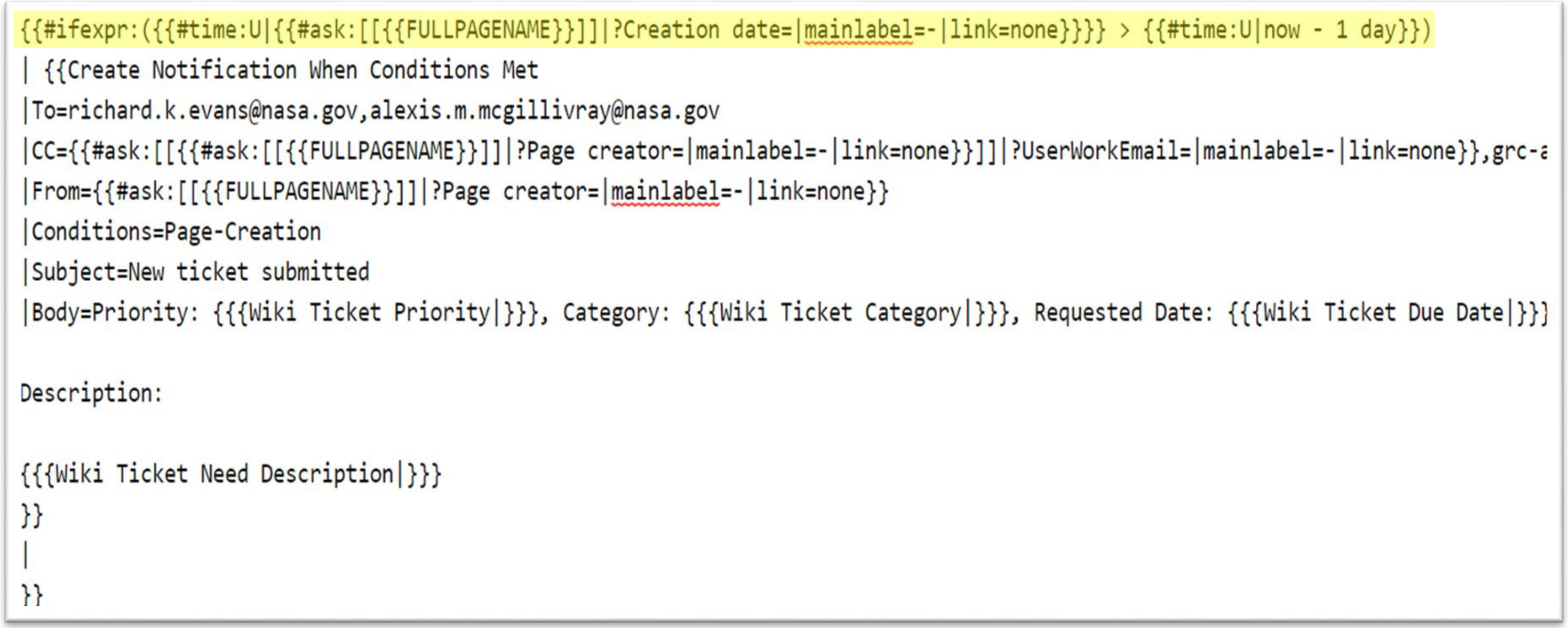
Design

Wiki Implementation

Let’s see an example in action. The following is a template call from our wiki ticketing tool.

The template which calls for the creation of an automatic notification is responsible for defining the logic around when the notification should be generated.

In this case, if the ticket was created today, we want to call **Template:Create Notification When Conditions Met** with the specified arguments to alert wiki admins of a new ticket in their inbox.



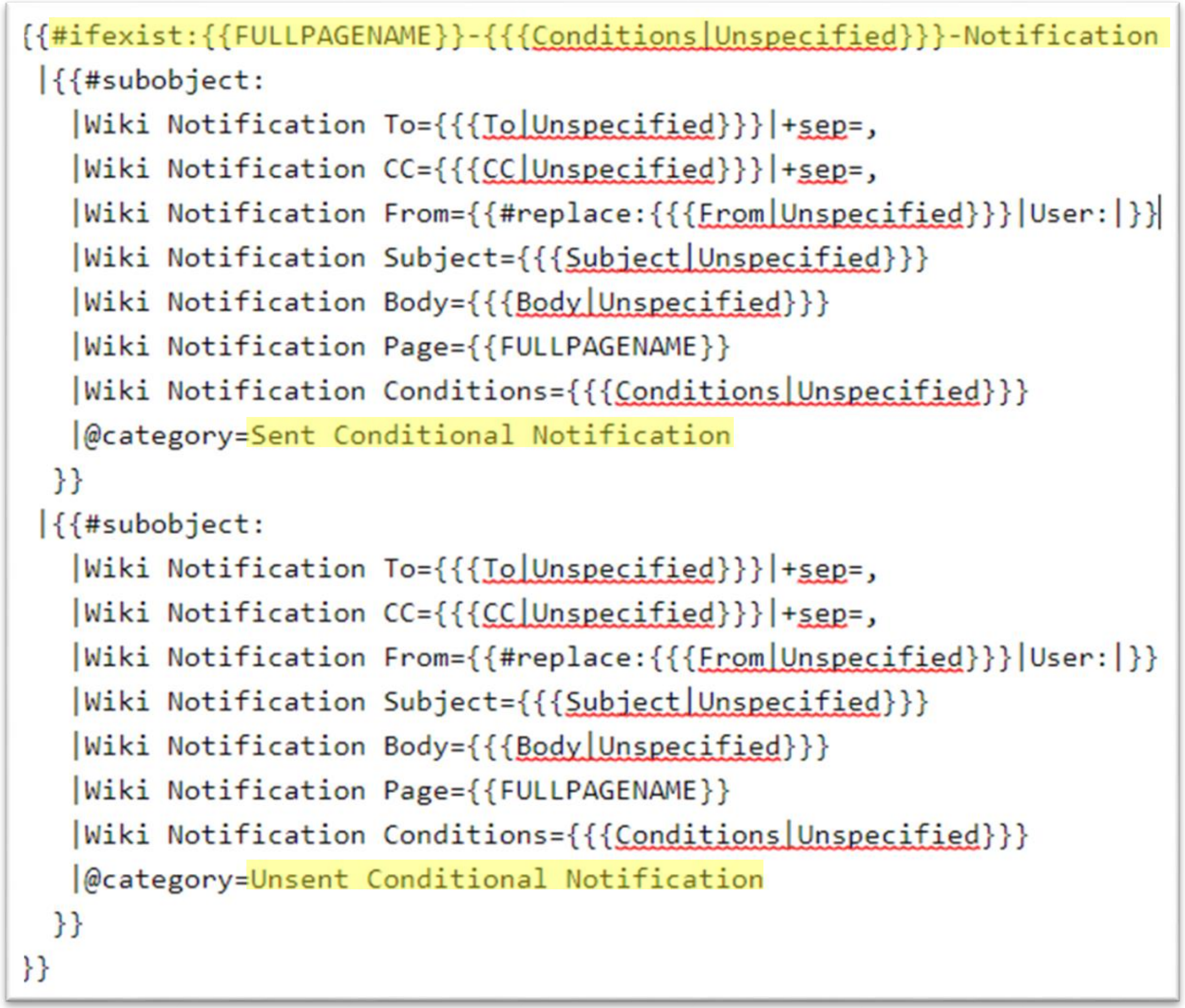
```
{{#ifexpr:({{#time:U|{{#ask:[[{{FULLPAGENAME}}]]|?Creation date=|mainlabel=-|link=none}}}} > {{#time:U|now - 1 day}})
| {{Create Notification When Conditions Met
|To=richard.k.evans@nasa.gov,alexis.m.mcgillivray@nasa.gov
|CC={{#ask:[[{{#ask:[[{{FULLPAGENAME}}]]|?Page creator=|mainlabel=-|link=none}}]]|?UserWorkEmail=|mainlabel=-|link=none}},grc-a
|From={{#ask:[[{{FULLPAGENAME}}]]|?Page creator=|mainlabel=-|link=none}}
|Conditions=Page-Creation
|Subject=New ticket submitted
|Body=Priority: {{{Wiki Ticket Priority|}}}, Category: {{{Wiki Ticket Category|}}}, Requested Date: {{{Wiki Ticket Due Date|}}}

Description:

{{{Wiki Ticket Need Description|}}}
}}
|
}}
```

Template:Wiki Ticket

Design

Wiki Implementation

If the notification has not been created yet, the sub-object will be in the **Unsent Conditional Notification** category. Else, it is in the **Sent Conditional Notification** category



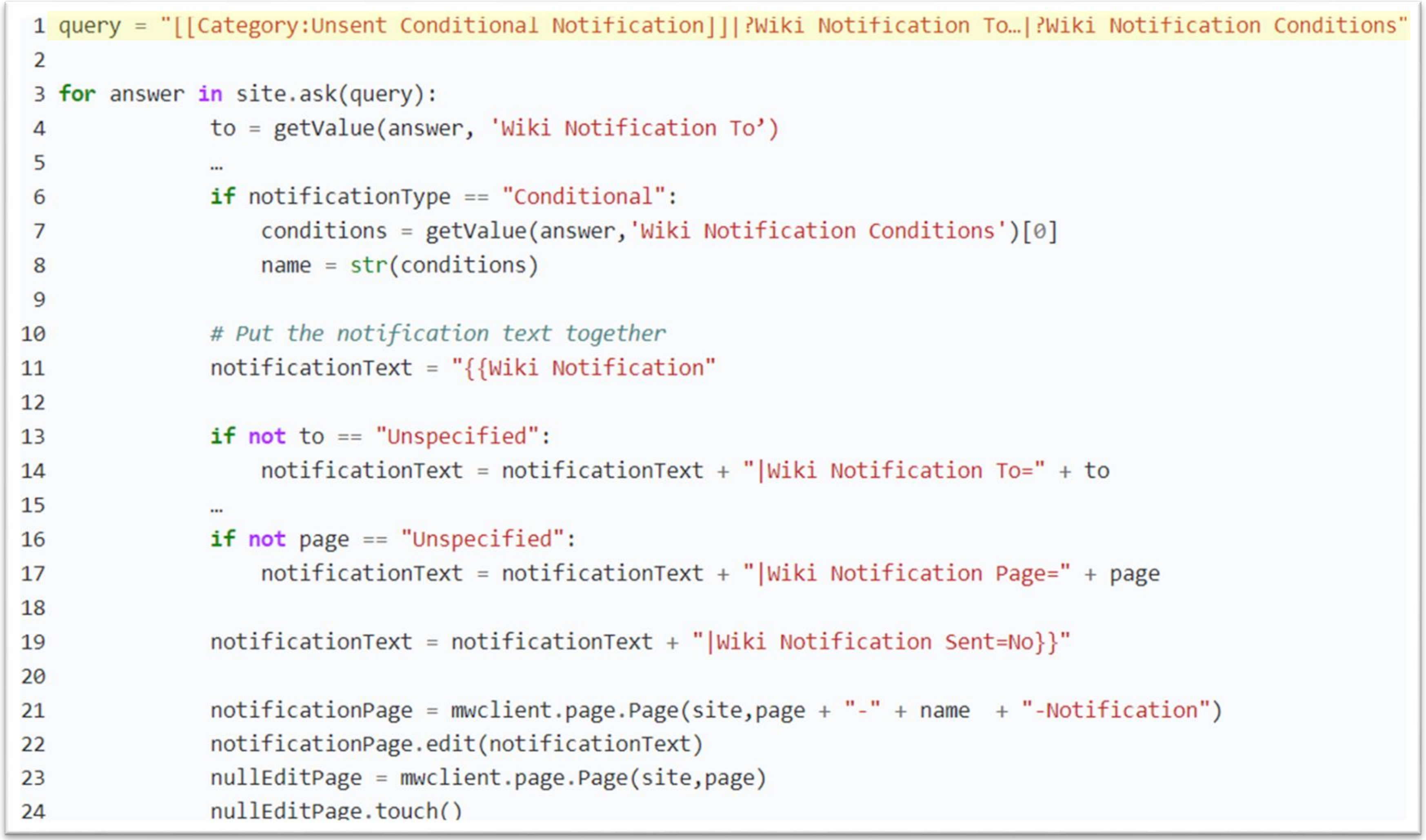
```
{{#ifexist:{{FULLPAGENAME}}-{{{Conditions|Unspecified}}}-Notification
 |{{#subobject:
   |Wiki Notification To={{{To|Unspecified}}}|+sep=,
   |Wiki Notification CC={{{CC|Unspecified}}}|+sep=,
   |Wiki Notification From={{#replace:{{{From|Unspecified}}}|User:|}}
   |Wiki Notification Subject={{{Subject|Unspecified}}}
   |Wiki Notification Body={{{Body|Unspecified}}}
   |Wiki Notification Page={{FULLPAGENAME}}
   |Wiki Notification Conditions={{{Conditions|Unspecified}}}
   |@category=Sent Conditional Notification
  }}
 |{{#subobject:
   |Wiki Notification To={{{To|Unspecified}}}|+sep=,
   |Wiki Notification CC={{{CC|Unspecified}}}|+sep=,
   |Wiki Notification From={{#replace:{{{From|Unspecified}}}|User:|}}
   |Wiki Notification Subject={{{Subject|Unspecified}}}
   |Wiki Notification Body={{{Body|Unspecified}}}
   |Wiki Notification Page={{FULLPAGENAME}}
   |Wiki Notification Conditions={{{Conditions|Unspecified}}}
   |@category=Unsent Conditional Notification
  }}
}}
```

Template:Create Notification When Conditions Met

Design

Server Implementation

- Queries are structured similarly to in-line #ask statements



```
query = "[[Category:Unsent Conditional Notification]]|?Wiki Notification To…|?Wiki Notification Conditions"

for answer in site.ask(query):
            to = getValue(answer, 'Wiki Notification To’)
            …
            if notificationType == "Conditional":
                conditions = getValue(answer,'Wiki Notification Conditions')[0]
                name = str(conditions)

            # Put the notification text together
            notificationText = "{{Wiki Notification"

            if not to == "Unspecified":
                notificationText = notificationText + "|Wiki Notification To=" + to
            …
            if not page == "Unspecified":
                notificationText = notificationText + "|Wiki Notification Page=" + page

            notificationText = notificationText + "|Wiki Notification Sent=No}}"

            notificationPage = mwclient.page.Page(site,page + "-" + name  + "-Notification")
            notificationPage.edit(notificationText)
            nullEditPage = mwclient.page.Page(site,page)
            nullEditPage.touch()
```

CreateNotificationPages.py (simplified)

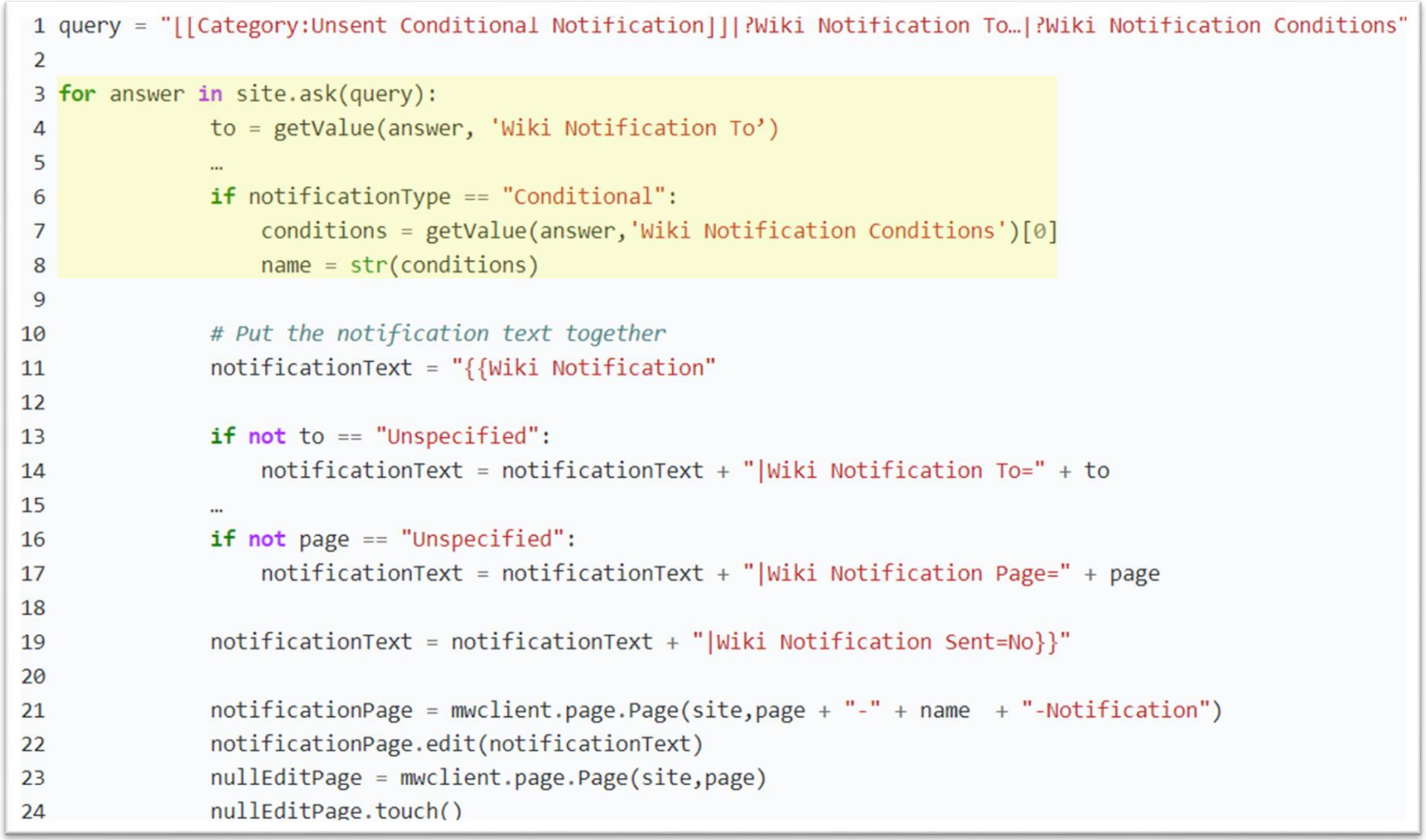
Note: NASA GRC-ATF uses python library mwclient, but this functionality can be accomplished through any script accessing the MediaWiki API

Design

Server Implementation

- Queries are structured similarly to in-line #ask statements

- The query response is returned as an iterable nested dictionary where the values can be assigned to variables



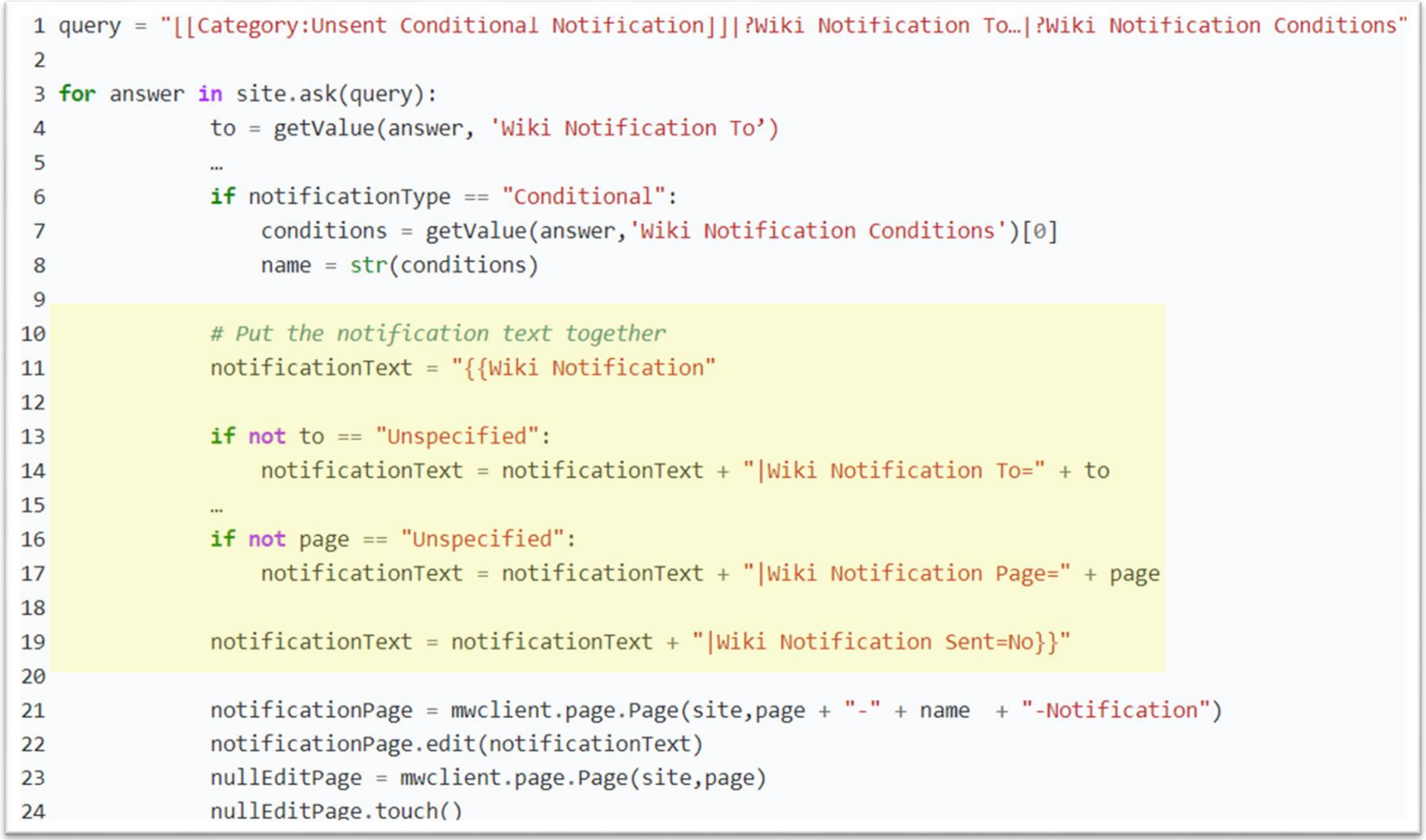
```
query = "[[Category:Unsent Conditional Notification]]|?Wiki Notification To…|?Wiki Notification Conditions"

for answer in site.ask(query):
            to = getValue(answer, 'Wiki Notification To’)
            …
            if notificationType == "Conditional":
                conditions = getValue(answer,'Wiki Notification Conditions')[0]
                name = str(conditions)

            # Put the notification text together
            notificationText = "{{Wiki Notification"

            if not to == "Unspecified":
                notificationText = notificationText + "|Wiki Notification To=" + to
            …
            if not page == "Unspecified":
                notificationText = notificationText + "|Wiki Notification Page=" + page

            notificationText = notificationText + "|Wiki Notification Sent=No}}"

            notificationPage = mwclient.page.Page(site,page + "-" + name  + "-Notification")
            notificationPage.edit(notificationText)
            nullEditPage = mwclient.page.Page(site,page)
            nullEditPage.touch()
```

CreateNotificationPages.py (simplified)

Design

Server Implementation

- Queries are structured similarly to in-line #ask statements
- The query response is returned as an iterable nested dictionary where the values can be assigned to variables
- The notification page content (template call and properties) are assembled



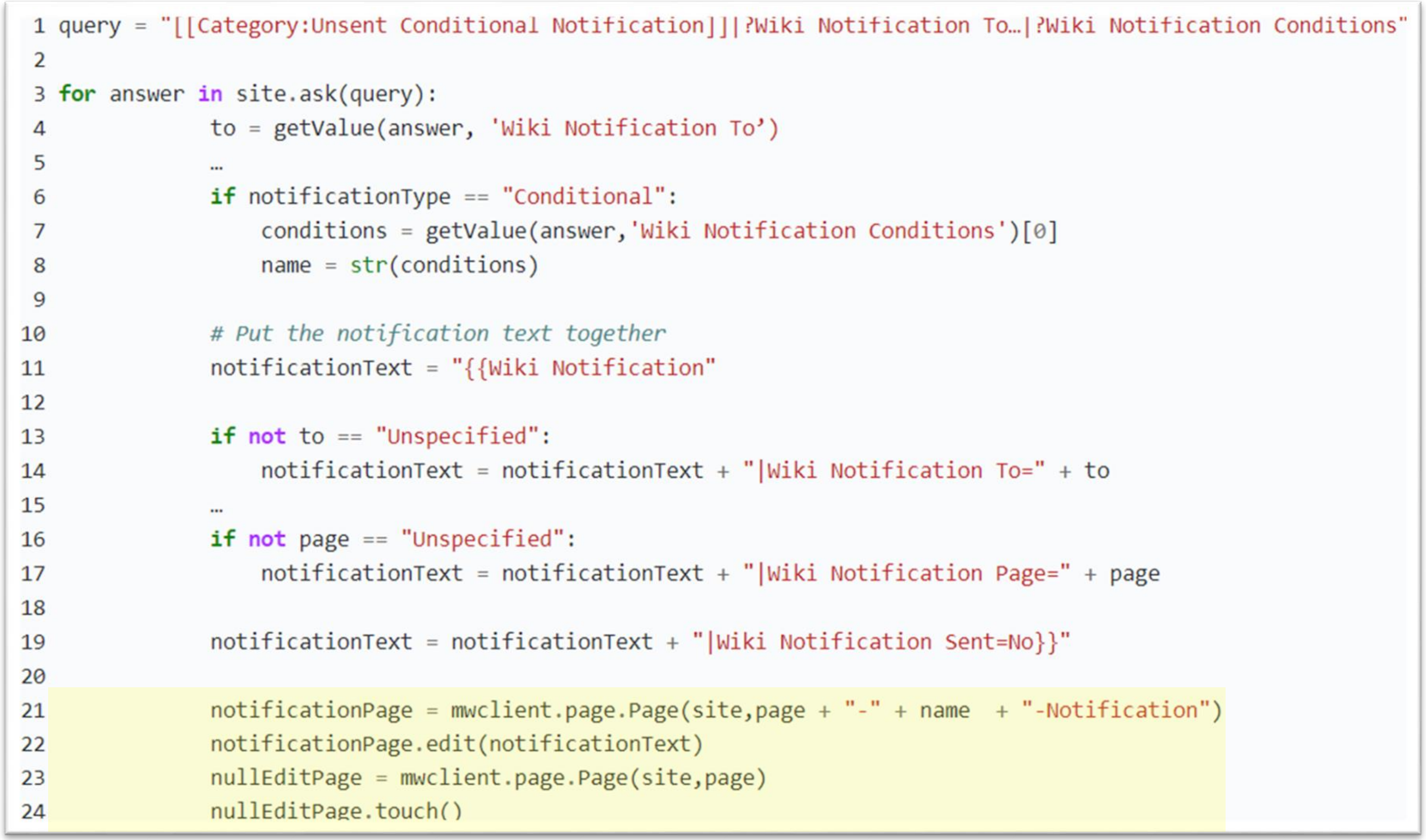
```
query = "[[Category:Unsent Conditional Notification]]|?Wiki Notification To…|?Wiki Notification Conditions"

for answer in site.ask(query):
            to = getValue(answer, 'Wiki Notification To’)
            …
            if notificationType == "Conditional":
                conditions = getValue(answer,'Wiki Notification Conditions')[0]
                name = str(conditions)

            # Put the notification text together
            notificationText = "{{Wiki Notification"

            if not to == "Unspecified":
                notificationText = notificationText + "|Wiki Notification To=" + to
            …
            if not page == "Unspecified":
                notificationText = notificationText + "|Wiki Notification Page=" + page

            notificationText = notificationText + "|Wiki Notification Sent=No}}"

            notificationPage = mwclient.page.Page(site,page + "-" + name  + "-Notification")
            notificationPage.edit(notificationText)
            nullEditPage = mwclient.page.Page(site,page)
            nullEditPage.touch()
```

CreateNotificationPages.py (simplified)

Design

Server Implementation

- Queries are structured similarly to in-line #ask statements
- The query response is returned as an iterable nested dictionary where the values can be assigned to variables
- The notification page content (template call and properties) are assembled
- The notification page is saved, and the page that caused its creation is null-edited so that the sub-object category is updated



```
query = "[[Category:Unsent Conditional Notification]]|?Wiki Notification To…|?Wiki Notification Conditions"

for answer in site.ask(query):
      to = getValue(answer, 'Wiki Notification To’)
      …
      if notificationType == "Conditional":
        conditions = getValue(answer,'Wiki Notification Conditions')[0]
        name = str(conditions)

      # Put the notification text together
      notificationText = "{{Wiki Notification"

      if not to == "Unspecified":
        notificationText = notificationText + "|Wiki Notification To=" + to
      …
      if not page == "Unspecified":
        notificationText = notificationText + "|Wiki Notification Page=" + page

      notificationText = notificationText + "|Wiki Notification Sent=No}}"

      notificationPage = mwclient.page.Page(site,page + "-" + name  + "-Notification")
      notificationPage.edit(notificationText)
      nullEditPage = mwclient.page.Page(site,page)
      nullEditPage.touch()
```

CreateNotificationPages.py (simplified)

Looking Forward

What’s the future of this work?

- We are planning on making our work wiki agnostic and integrating it into Project Open CSP
- We are in the process of making all of the GRC-ATF KM tools and scripts available on the public NASA github

Questions?

alexis.m.mcgillivray@nasa.gov