

UC07 – Determine and Deliver ERR Benefits Use Case Specification  
**State of MN - eHEAT Phase III**  
Version 1.3





## Revision History

Date	Version	Description	Author
02/10/2004	DRAFT	Initial Version	Ramu Srinivasan
02/17/2004	1.0	Peer Review with Narayan Srinivasan and submitted for sign off	Ramu Srinivasan
02/27/2004	1.1	Updated based on Sign off comment responses. Updated business rules for selecting application to perform ERR.	Ramu Srinivasan
05/26/2004	1.2	Removed 'No Fund' payment status due to design consideration changes. No fund is a flag and not an status any more.	Ramu Srinivasan
08/09/2004	1.3	Updated with Process Communication	Ramu Srinivasan



## Table of Contents

1.	<u>Brief Description</u> .....	4
2.	<u>Actors</u> .....	4
3.	<u>Activity Diagram</u> .....	5
4.	<u>Flow of Events</u> .....	6
	4.1 <u>Basic Flow</u> .....	6
	4.2 <u>Alternate Flow</u> .....	9
5.	<u>Special Requirements</u> .....	9
6.	<u>Preconditions</u> .....	9
7.	<u>Post conditions</u> .....	9
8.	<u>Business Rules</u> .....	9
	<u>Security</u> .....	9
	<u>ERR Event</u> .....	10
	<u>ERR Task Maintenance</u> .....	10



## Determine and Deliver ERR Benefits

### 1. Brief Description

This use case provides a way to Determine and Deliver ERR Benefits.

### 2. Actors

Service Provider, DOC User

Space below is intentionally left blank

### 3. Activity Diagram

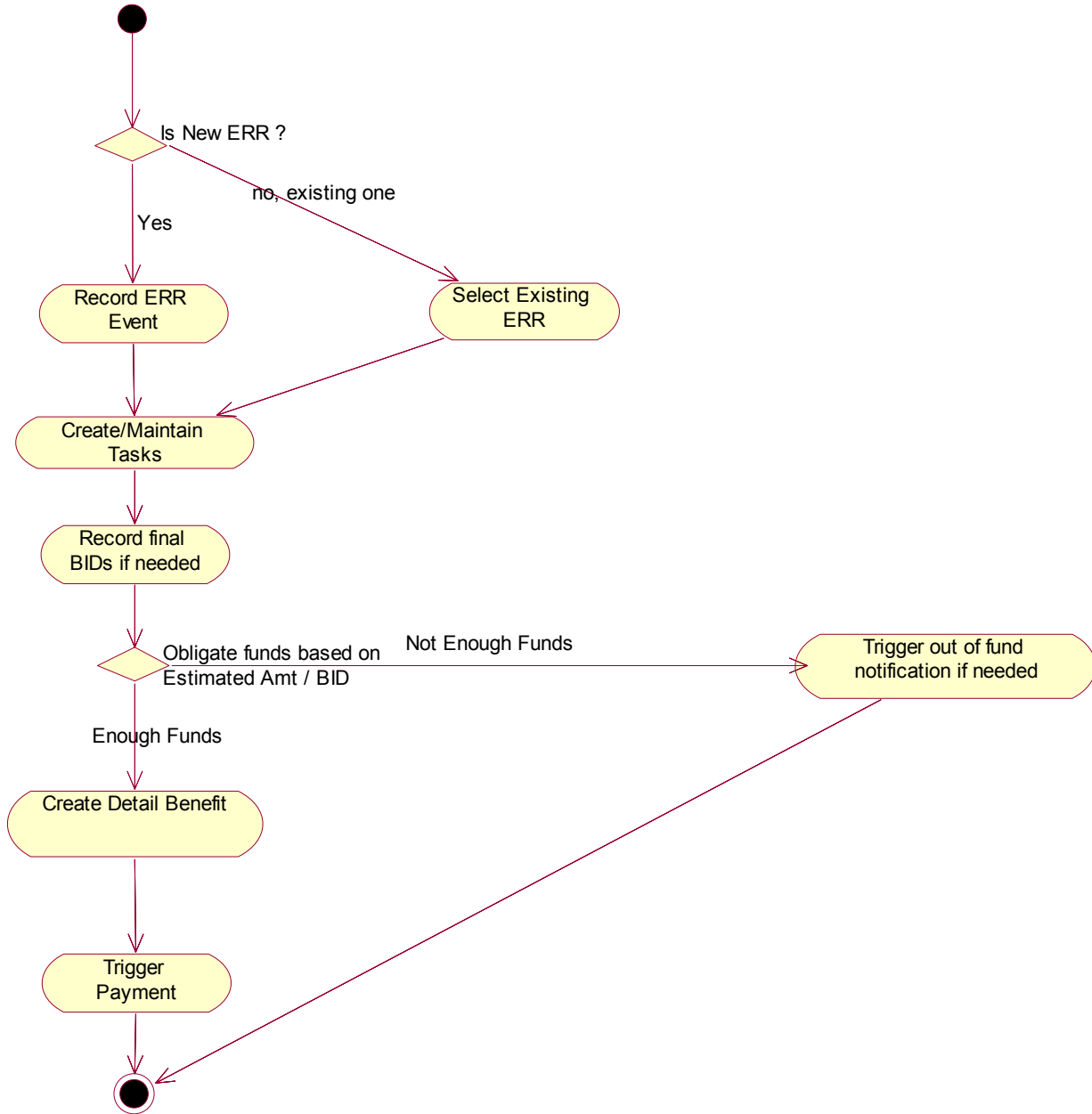


Figure 1: Determine and Deliver ERR Benefits Activity Diagram

## 4 Flow of Events

### 4.1 Basic Flow

#### 4.1.1 Start of the Use Case

This use case starts when user clicks “Search by Application” from the ERR Benefits tab from the Client Services menu.

#### 4.1.2 Selecting by Application

1. User enters search criteria and selects an application from the search results.
2. On clicking “Maintain ERR Event” button, system displays a history of ERR events for selected application if any.
3. The user can either select an existing ERR event to modify or create a new ERR event.

#### 4.1.3 Determine and Deliver ERR Benefits

1. User enters relevant information in event screen.
1. An event can have zero to many tasks. Tasks can be maintained (added/edited) by clicking on “Maintain Tasks”.
2. The system keeps track of event statuses and a history of status each time when it changes along with date. The following table lists event statuses and the source that drives it.

Event Status	Source Driven	When / Rules
No Action	System	<ul style="list-style-type: none"> <li>• First time when an event is created and no action is taken.</li> <li>• This implies that no tasks have been created / added.</li> </ul>
In Progress	System	<ul style="list-style-type: none"> <li>• When a new task is added to this event for the first time</li> </ul>
Complete	System	<ul style="list-style-type: none"> <li>• When all the tasks for the event are either complete or cancelled.</li> </ul>
Denied	User	<ul style="list-style-type: none"> <li>• When user decides this event needs to be denied.</li> <li>• An event can be denied only when tasks status of that event are either “Cancelled” or “In Progress”.</li> <li>• Any tasks with obligated fund</li> </ul>



## State of Minnesota – eHEAT Phase III



		<p>for that event will be de-obligated.</p> <ul style="list-style-type: none"> <li>• A reason for denial can be added to “Notes” field.</li> </ul>
--	--	--

3. The user at this point has two options. a) Save the information by clicking on “Save” and come back later to change information / add tasks. b) Add tasks associated with this event.
4. Clicking on “Maintain tasks” will automatically save any changes made to event related information.
5. Relevant information will be entered by the user for each task. More tasks can be added by clicking on “Add More Tasks”. Tasks can be removed from the screen before it is saved by clicking “Remove”. But once saved, it can’t be removed.
6. User can click “Save” button to save information. This will also obligate funds if accounting fund was used and if fund wasn’t previously obligated. Upon successful save/obligation, system will display an appropriate message like “Information saved and Funds for \$xxx were successfully obligated” in the message bar of the same screen. In the event of failure such as insufficient funds, system will display an appropriate message indicating a reason of failure.
7. Obligate button can also be used to obligate funds. If previously there was no fund and now new funds have arrived, user can click “Obligate”.
8. No Fund for each task is identified by ‘No fund’ flag and this flag is set when there are no funds during first time obligation.
9. The following table lists possible values for task status, source that drives and when it can happen. System will also maintain a history of task status changes and date when it was changed.

Task Status	Source Driven	When / Rules
In Progress	User/System	<ul style="list-style-type: none"> <li>• When a new task is added for the first time</li> <li>• When user decides to resume a task which was previously on hold</li> <li>• When a user decides that a task is not complete (may be after inspection) though the task status was changed to “Complete”.</li> </ul>
On Hold	User	<ul style="list-style-type: none"> <li>• When user decides to put this on hold for some reason. Fund if obligated remains obligated.</li> </ul>
Complete	User	<ul style="list-style-type: none"> <li>• When user decides that this task is complete.</li> </ul>
Cancelled	User	<ul style="list-style-type: none"> <li>• When user decides this task needs to be cancelled or</li> </ul>

		deleted.
--	--	----------

10. Payment status is tied to its task’s status and can be one of the following values. It also describes the source that drives it and when it can occur. Like other statuses, histories of payment status changes are also maintained along with date indicating when it was changed.

Payment Status	Source Driven	When / Rules
N/a	System	<ul style="list-style-type: none"> <li>When a new task is added for the first time</li> <li>When a non-accounting fund is used.</li> </ul>
Obligated	System	<ul style="list-style-type: none"> <li>When user uses an accounting fund for one or more tasks and funds need to be set a side.</li> </ul>
Certifiable	System	<ul style="list-style-type: none"> <li>When a task is marked as “Complete” indicating that this transaction may be ready for payment certification.</li> </ul>
Payable	System	<ul style="list-style-type: none"> <li>When user certifies/approves a payment transaction.</li> </ul>
Payment In Progress	System	<ul style="list-style-type: none"> <li>When Payment process has picked up certified payment transactions.</li> </ul>
Paid	System	<ul style="list-style-type: none"> <li>When payment has been sent to client/vendor/contractor.</li> </ul>

11. When tasks are created and saved, system will try to obligate accounting fund and will display a message indicating successful obligation or no fund availability.
12. When estimated/bid Amount or actual amount is changed, the user can re-obligate by hitting “Obligate” button or “Save” Button. Appropriate message will be displayed upon success or failure of obligation.
13. Once one or more tasks are complete, user can enter inspection report by clicking on “Maintain Post ERR Inspection”.
14. User can enter or edit an existing inspection report by selecting “Edit” or “Add” respectively from Inspection List Screen.
15. System will keep track of dates when inspection was added.

### 4.1.3 Use Case Ends

The use case ends on display of the message.



---

## 4.2 Alternate Flow

### 4.2.1 Selecting by ERR Event

1. User clicks “Search by ERR Event” from ERR Benefits tab from Client Services menu.
2. After user enters search criteria the system displays a result based on search criteria.
3. User selects an ERR event and clicks “Maintain ERR Event”.
4. Alternate flow ends here and continues with normal flow 4.1.3.

## 5 Special Requirements

- Contractor Maintenance screen that will allow service providers to maintain their own list of contractor ids and contractor names with a flag to indicate “Active” / “In Active” status.
- A Fund table containing list of funds that will be used for ERR with a flag to indicate accounting/non-accounting fund, another flag to indicate whether transactions using that funds source will be paid centrally.

## 6 Preconditions

Applications status should be complete and approved.

## 7 Post conditions

An ERR Event will be created and can have 0+ tasks associated with it.

## 8 Business Rules

### Security:

- State user has read only access to all the transaction screens. In other words state user will not be able to create or modify ERR events and tasks. In addition, state user will have a privilege to select any service provider/agency from the agency drop down list in application search.
- Authorized Service providers will have read/write access to all ERR screens and will be able to create, modify events and tasks.
- All other service provider users will have read only access to all the transaction screens. In other words these service provider users will not be able to create or modify ERR events and tasks.

### **ERR Event:**

- An ERR event can be created for completed and approved applications.
- An ERR event can be saved without any task.
- Reported and response date/time will be used to track time taken to respond to an ERR event and must be within past 7 days from the current system date/time. It cannot have future date/time.
- Response date/time should be greater than or equal to reported date.
- An event can be “Denied” either if there are no tasks attached to this event or if statuses of all the tasks that are attached to this event are cancelled.
- Edits can be made to any fields in the event as long as event’s status is not “Complete” or “Denied”.

### **ERR Task Maintenance:**

#### Tasks:

- Estimate/Bid Amount and Final Amount should be a positive value and may contain 0.
- Having 0 for estimated/Bid Amount field will allow all the tasks of a contractor to be listed in an event while contractor has only one estimate for all of his tasks.

#### Funds:

- Non-accounting Fund “Referred” will be used as alternative solution for providing restricted access to fund source dropdown list for peoples in same agency.

#### Contractor list:

- Payments cannot be made directly to household.
- Only active contractors will be displayed in drop down box. This applies to new tasks.
- For tasks, which had active contractors before, but then became inactive before task completion, those contractors will still be listed in the drop down box. User will be permitted to change to a new active contractor, but will not be allowed to select an inactive contractor for existing/new tasks.

#### Obligation:

- The payment status becomes obligated when there are sufficient funds. Obligation is done only for accounting funds.
- When there is no fund, system will generate out of fund error and will be displayed on the screen. No Fund for each task is identified by ‘No fund’ flag and this flag is set when there are no funds during first time obligation.



- If funds were previously obligated and if there is an attempt to change estimated/bid amount or the actual amount, the user needs to click “Obligate” button or “Save” button in order for system to re-obligate and the system will save the changes only if obligation was successful. If unsuccessful, system will not save changes and will remain as it was before re-obligation.

### Task Completion:

- Rule will be added so that the status of a task can not be changed to complete when “Type” is replacement and is not inspected (i.e. inspected check box is not checked).
- An ERR task that is assigned to service provider’s crew, when completed, the payment status will be set to ‘Paid’. This will allow payments for crew to be put to a terminal state as crew get paid thru payroll and do not receive checks from eHEAT system.

### Payments:

- If different fund categories need to be used for same task/contractor, then multiple tasks can be created for same contractor with split amount for that one task and each of them using different fund categories.
- A rule will be imposed such that payment process will pick all the tasks of a contractor only if they are complete. This will prevent payments to contractor until all of his work is done. This rule is only within a contractor and not across contractor. So, a contractor doing mechanical work can get paid when his task is complete while building shell contractor’s work is still in progress for that event.
- If same contractor needs to get paid for one or more tasks before all of his tasks are complete, then a separate event needs to be created.

### Others:

- System maintains date created and date modified for all events and tasks.
- Tasks are editable as long as payment status is not “Payment in progress” or “Paid”.
- When the user clicks “Quit” button, the system will take the user to previous screen or search screen (if any in that section) or home page of that tab which ever is appropriate. Clicking “Quit” button will not save any changes made (provided they haven’t been saved before) and will ask for confirmation before proceeding further.