Analysis of Bridges & Wormholes Paper

Part 2 – Assessment and Improvement of <u>Control</u>

First Delivered: 03 Feb 2026

Last Updated: 09 Feb 2026

Michael M Lee

Introduction

Purpose of Analysis Effort

- ✓ Understand the wormhole approach to bridging (1st control, then data)
- ✓ Provide an overview of the approach that makes it more accessible
- Assess if the responsibilities of the SW Architecture proposed in the B&W paper support the behavior of bridges as presented in Part 1.
- Identify constraints and propose improvements

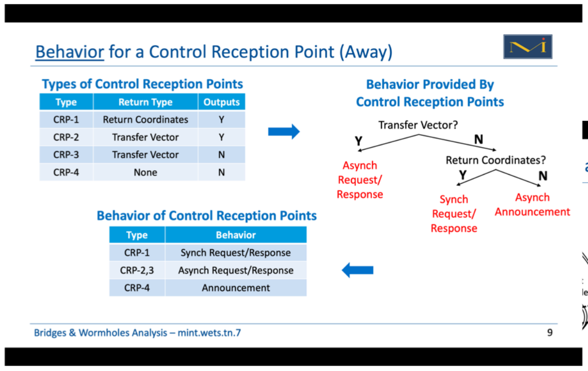
Key Terms

- HOME (requesting service) domain
- AWAY (providing service) domain
- MX = Model Execution/Software Architecture domain

When We Last Visited… Bridging Behaviors

Bridging Behaviors

Do the SW Architecture responsibilities support them?



Behavior for a Control Reception Point (Away)

Types of Control Reception Points

Type	Return Type	Outputs
CRP-1	Return Coordinates	Y
CRP-2	Transfer Vector	Y
CRP-3	Transfer Vector	N
CRP-4	None	N

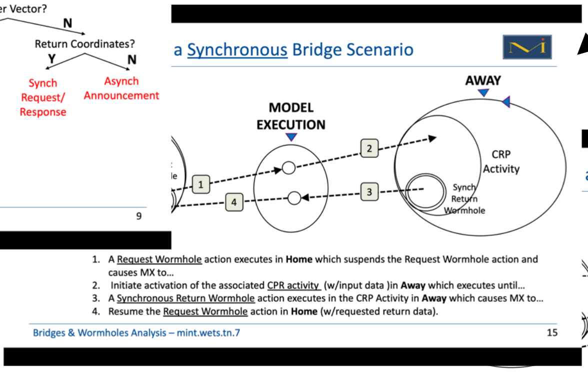
Behavior Provided By Control Reception Points

Behavior of Control Reception Points

Type	Behavior
CRP-1	Synch Request/Response
CRP-2,3	Asynch Request/Response
CRP-4	Announcement

Bridges & Wormholes Analysis – mint.wets.tn.7 9

CRP Behavior

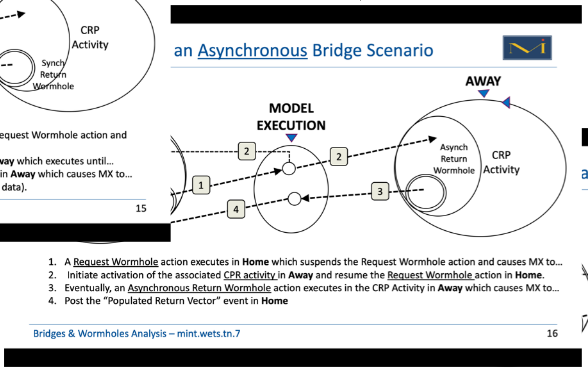


a Synchronous Bridge Scenario

1. A Request Wormhole action executes in **Home** which suspends the Request Wormhole action and causes MX to…
2. Initiate activation of the associated CPR activity (w/input data)in **Away** which executes until…
3. A Synchronous Return Wormhole action executes in the CRP Activity in **Away** which causes MX to…
4. Resume the Request Wormhole action in **Home** (w/requested return data).

Bridges & Wormholes Analysis – mint.wets.tn.7 15

Synchronous Bridge Scenario

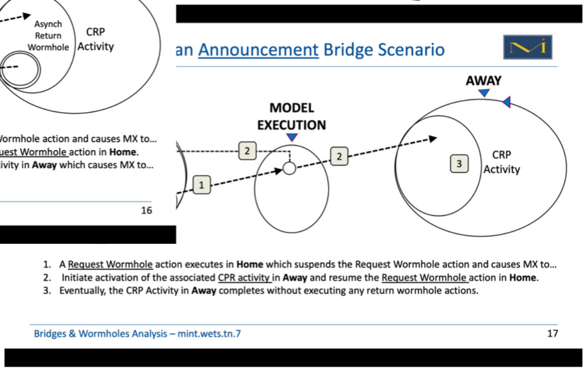


an Asynchronous Bridge Scenario

1. A Request Wormhole action executes in **Home** which suspends the Request Wormhole action and causes MX to…
2. Initiate activation of the associated CPR activity in **Away** and resume the Request Wormhole action in **Home**.
3. Eventually, an Asynchronous Return Wormhole action executes in the CRP Activity in **Away** which causes MX to…
4. Post the “Populated Return Vector” event in **Home**

Bridges & Wormholes Analysis – mint.wets.tn.7 16

Asynchronous Bridge Scenario



an Announcement Bridge Scenario

1. A Request Wormhole action executes in **Home** which suspends the Request Wormhole action and causes MX to…
2. Initiate activation of the associated CPR activity in **Away** and resume the Request Wormhole action in **Home**.
3. Eventually, the CRP Activity in **Away** completes without executing any return wormhole actions.

Bridges & Wormholes Analysis – mint.wets.tn.7 17

Announcement Bridge Scenario

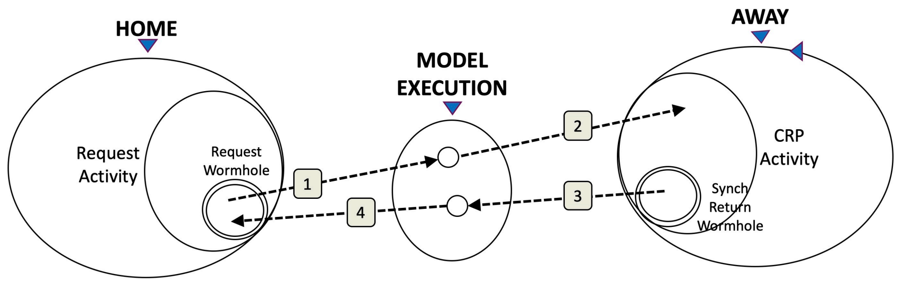
B&W Allocation of Responsibility to SW Architecture

Item	Type	Page	Paragraph	Section	SW Architecture Responsibility
1	Data	3	2	Assumptions & Rationale	Establish instance correspondence across domains
2	Data	3	3	Assumptions & Rationale	Support semantic shifts
3	Control	3	4	Transfer Vectors	Structure/Type of Transfer Vector
4	Control	4	4	Transfer Vectors	Construction, population & propagation of a return event
5	Control	5	2	Return Coordinates…	Structure/Type of Return Coordinates
6	Control	5	5	Return Coordinates…	Manage fork in control between Home & Away
7	Control	5	8	Return Coordinates…	Construction & population of Return Coordinates
8	Control	6	1	Domain Interface (CRP)	*Transfer control to Away via an event or invocation
9	Control	7	6	Request Wormhole	*Transfer control from Home to Away via a Request Action
10	Control	9	4	Return Wormhole	*Transfer control from Away to Home via a Return Action with a Transfer Vector or Return Coordinates
11	Control	14	3	Resolving Wormholes	Construction & population of a return event
					* = Responsibility implied by wormholes being MX actions

References for Implied Responsibilities

Item	Type	Page	Paragraph	Section	SW Architecture Responsibility	Bridges & Wormholes Reference
8	Control	6	1	Domain Interface (CRP)	*Transfer control to Away via an event or invocation	"When a domain receives control from across a domain boundary, the receiver sees the control as either the arrival of an event from outside the domain or as the invocation of a synchronous service."
9	Control	7	6	Request Wormhole	*Transfer control from Home to Away via a Request Action	"Control is transferred from one domain to another by invocation of a wormhole process on an ADFD or SDFD."
10	Control	9	4	Return Wormhole	*Transfer control from Away to Home via a Return Action with a Transfer Vector or Return Coordinates	* "If there is an input to the wormhole of type transfer vector, the return will be asynchronous." * "If there is an input of type return coordinate, the return will be synchronous."

Support for a Synchronous Bridge Scenario?



Item	Type	Page	Paragraph	Section	SW Architecture Responsibility
1	Data	3	2	Assumptions & Rationale	Establish instance correspondence across domains
2	Data	3	3	Assumptions & Rationale	Support semantic shifts
3	Control	3	4	Transfer Vectors	Structure/Type of Transfer Vector
4	Control	4	4	Transfer Vectors	Construction, population & propagation of a return event
5	Control	5	2	Return Coordinates…	Structure/Type of Return Coordinates
6	Control	5	5	Return Coordinates…	Manage fork in control between Home & Away
7	Control	5	8	Return Coordinates…	Construction & population of Return Coordinates
8	Control	6	1	Domain Interface (CRP)	*Transfer control to Away via an event or invocation
9	Control	7	6	Request Wormhole	*Transfer control from Home to Away via a Request Action
10	Control	9	4	Return Wormhole	*Transfer control from Away to Home via a Return Action with a Transfer Vector or Return Coordinates
11	Control	14	3	Resolving Wormholes	Construction & population of a return event
					* = Responsibility implied by wormholes being MX actions

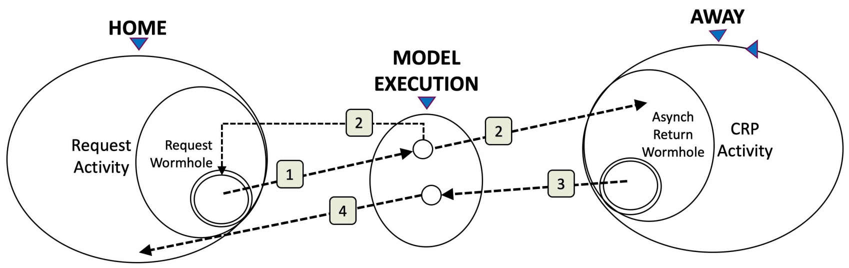
How Scenario Works

1. A Request Wormhole action executes in **Home** which passes control to MX [#9] which then…
2. Suspends the Request Wormhole Action [#6], creates a Return Coordinate [#5, #7] and initiates activation of the associated CRP activity in **Away** [#8] which stores the Return Coordinate and executes until…
3. A Synchronous Return Wormhole action executes in the CRP Activity [#10] in **Away** which has retrieved and passed the Return Coordinate and any return data to MX which causes it to…
4. Resume the Request Wormhole action in **Home** [#6].

Supported?

Behavior is consistent with B&W paper's published SW Architecture responsibilities.

Support an Asynchronous Bridge Scenario?



Item	Type	Page	Paragraph	Section	SW Architecture Responsibility
1	Data	3	2	Assumptions & Rationale	Establish instance correspondence across domains
2	Data	3	3	Assumptions & Rationale	Support semantic shifts
3	Control	3	4	Transfer Vectors	Structure/Type of Transfer Vector
4	Control	4	4	Transfer Vectors	Construction, population & propagation of a return event
5	Control	5	2	Return Coordinates…	Structure/Type of Return Coordinates
6	Control	5	5	Return Coordinates…	Manage fork in control between Home & Away
7	Control	5	8	Return Coordinates…	Construction & population of Return Coordinates
8	Control	6	1	Domain Interface (CRP)	*Transfer control to Away via an event or invocation
9	Control	7	6	Request Wormhole	*Transfer control from Home to Away via a Request Action
10	Control	9	4	Return Wormhole	*Transfer control from Away to Home via a Return Action with a Transfer Vector or Return Coordinates
11	Control	14	3	Resolving Wormholes	Construction & population of a return event
					* = Responsibility implied by wormholes being MX actions

How Scenario Works

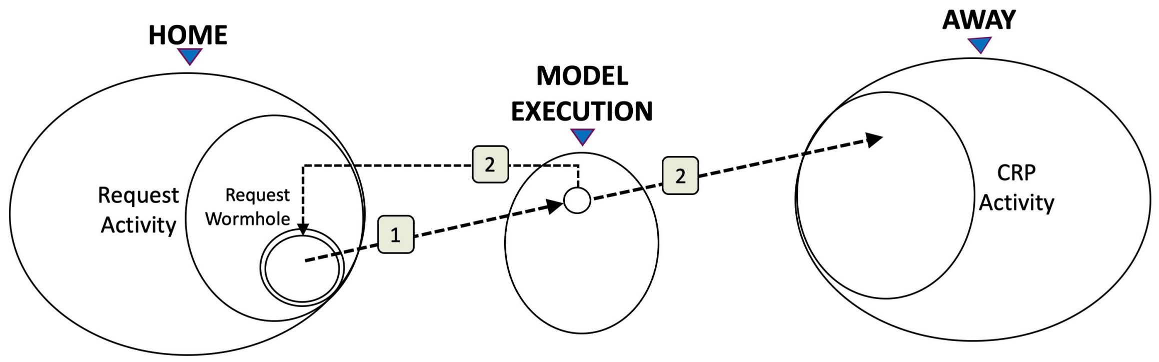
1. A Request Wormhole action executes in **Home** which passes control to MX [#9] which then…
2. Returns control to the Request Wormhole action [OOA] and initiates activation of the associated CRP activity in **Away** [#8] which stores the Transfer Vector and executes until…
3. An Asynchronous Return Wormhole action executes in the CRP Activity [#10] in **Away** which retrieves and passes the Transfer Vector and any return data to MX which causes it to…
4. Construct a “Populated Return Vector” event [#3, #4] and post that event in **Home.** [#11, #4]

Support?

Behavior is consistent with B&W paper’s published SW Architecture responsibilities.

Support for an <u>Announcement</u> Bridge Scenario?

Announcement Bridge scenario is just a subset of the Asynchronous Bridge scenario.



Supported?

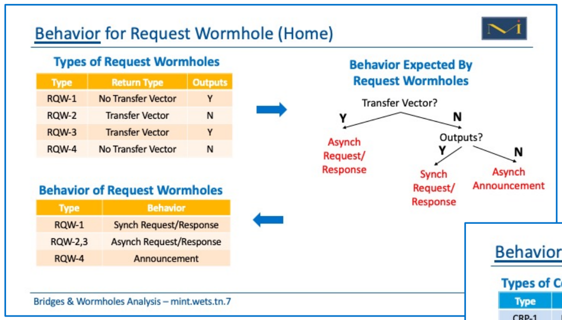
Behavior is consistent with B&W paper's published SW Architecture responsibilities.

.

Control Responsibilities Used in Scenarios

Item	Type	Page	Paragraph	Section	SW Architecture Responsibility
1	Data	3	2	Assumptions & Rationale	Establish instance correspondence across domains
2	Data	3	3	Assumptions & Rationale	Support semantic shifts
✓ 3	Control	3	4	Transfer Vectors	Structure/Type of Transfer Vector
✓ 4	Control	4	4	Transfer Vectors	Construction, population & propagation of a return event
✓ 5	Control	5	2	Return Coordinates…	Structure/Type of Return Coordinates
✓ 6	Control	5	5	Return Coordinates…	Manage fork in control between Home & Away
✓ 7	Control	5	8	Return Coordinates…	Construction & population of Return Coordinates
✓ 8	Control	6	1	Domain Interface (CRP)	*Transfer control to Away via an event or invocation
✓ 9	Control	7	6	Request Wormhole	*Transfer control from Home to Away via a Request Action
✓ 10	Control	9	4	Return Wormhole	*Transfer control from Away to Home via a Return Action with a Transfer Vector or Return Coordinates
✓ 11	Control	14	3	Resolving Wormholes	Construction & population of a return event
					* = Responsibility implied by wormholes being MX actions

When We Last Visited… Matching Requests and CRPs



Behavior for Request Wormhole (Home)

Types of Request Wormholes

Type	Return Type	Outputs
RQW-1	No Transfer Vector	Y
RQW-2	Transfer Vector	N
RQW-3	Transfer Vector	Y
RQW-4	No Transfer Vector	N

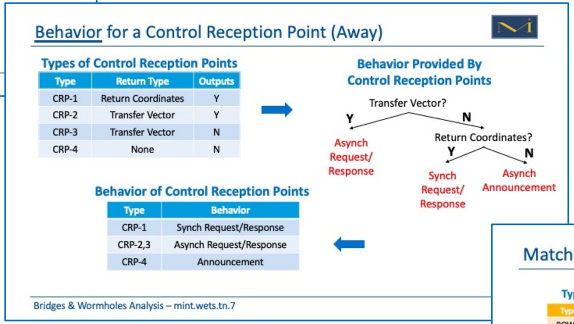
Behavior Expected By Request Wormholes

Behavior of Request Wormholes

Type	Behavior
RQW-1	Synch Request/Response
RQW-2,3	Asynch Request/Response
RQW-4	Announcement

Bridges & Wormholes Analysis – mint.wets.tn.7

Request Behavior



Behavior for a Control Reception Point (Away)

Types of Control Reception Points

Type	Return Type	Outputs
CRP-1	Return Coordinates	Y
CRP-2	Transfer Vector	Y
CRP-3	Transfer Vector	N
CRP-4	None	N

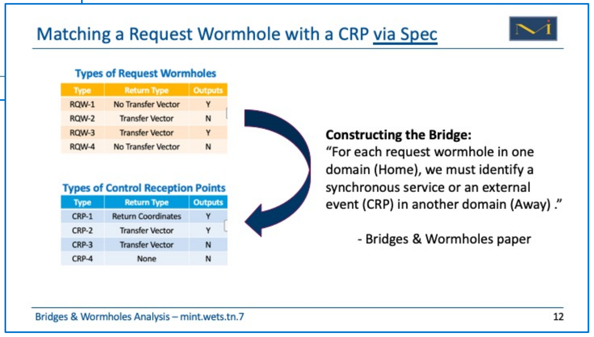
Behavior Provided By Control Reception Points

Behavior of Control Reception Points

Type	Behavior
CRP-1	Synch Request/Response
CRP-2,3	Asynch Request/Response
CRP-4	Announcement

Bridges & Wormholes Analysis – mint.wets.tn.7

CRP Behavior



Matching a Request Wormhole with a CRP via Spec

Types of Request Wormholes

Type	Return Type	Outputs
RQW-1	No Transfer Vector	Y
RQW-2	Transfer Vector	N
RQW-3	Transfer Vector	Y
RQW-4	No Transfer Vector	N

Types of Control Reception Points

Type	Return Type	Outputs
CRP-1	Return Coordinates	Y
CRP-2	Transfer Vector	Y
CRP-3	Transfer Vector	N
CRP-4	None	N

Constructing the Bridge:
"For each request wormhole in one domain (Home), we must identify a synchronous service or an external event (CRP) in another domain (Away) ."

- Bridges & Wormholes paper

Bridges & Wormholes Analysis – mint.wets.tn.7 12

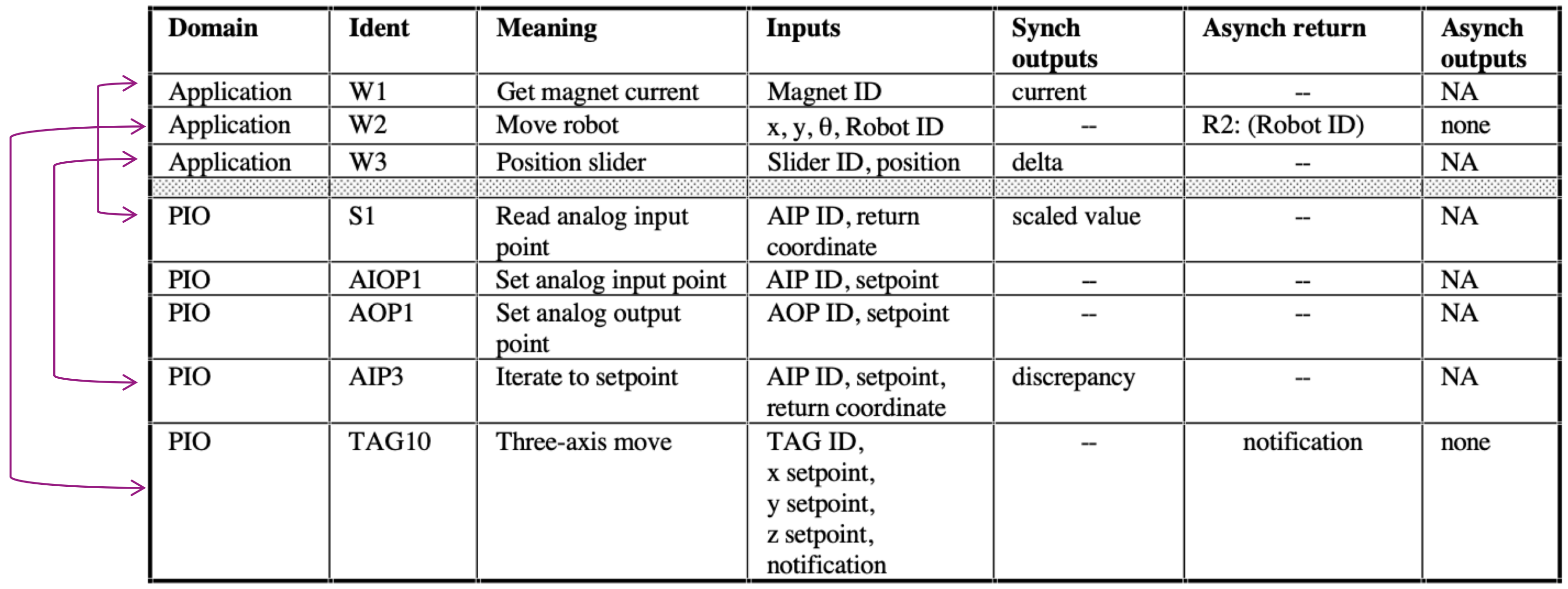
Matching Request with CRP

What are the constraints on matching?

Can these constraints be reduced?

Matching Wormholes with Control Reception Points

In order to identify the CRP that corresponds to each wormhole, it is helpful to bring the definitions of all request wormholes from one domain together with all CRPs of the other domain. One way of doing this is shown in the following table:



Domain	Ident	Meaning	Inputs	Synch outputs	Asynch return	Asynch outputs
Application	W1	Get magnet current	Magnet ID	current	--	NA
Application	W2	Move robot	x, y, θ, Robot ID	--	R2: (Robot ID)	none
Application	W3	Position slider	Slider ID, position	delta	--	NA
PIO	S1	Read analog input point	AIP ID, return coordinate	scaled value	--	NA
PIO	AIOP1	Set analog input point	AIP ID, setpoint	--	--	NA
PIO	AOP1	Set analog output point	AOP ID, setpoint	--	--	NA
PIO	AIP3	Iterate to setpoint	AIP ID, setpoint, return coordinate	discrepancy	--	NA
PIO	TAG10	Three-axis move	TAG ID, x setpoint, y setpoint, z setpoint, notification	--	notification	none

Constraints on Matching Behavior w/B&W v1

HOME Request
w/o Transfer Vector
and w/outputs
(Sync)

Only Matches

AWAY CRP
w/Return Coordinates
and Sync Return
(Sync)

HOME Request
w/Transfer Vector
w/ or w/o outputs
(Async)

Only Matches

AWAY CRP
w/Transfer Vector
and Async Return
(Async)

HOME Request
w/o Transfer Vector
and w/o outputs
(Announce)

Only Matches

AWAY CRP
w/o Transfer Vector
or Return Coordinates
And w/o any Return
(Announce)

NOTE: I/O Data must also match

Constraints and Proposed Improvements

Bridges & Wormholes v1

- 1 Request Wormhole to address:
  - synchronous,
  - asynchronous and
  - announcement behavior
- 2 Return Wormholes to address:
  - synchronous and
  - asynchronous behavior

Constraints

- A CRP can only provide 1 behavior based on Return Wormhole it uses
- Home must match CRP behavior
- Hard to determine Home's desired behavior from the Request in AL (Transfer Vector & Outputs)

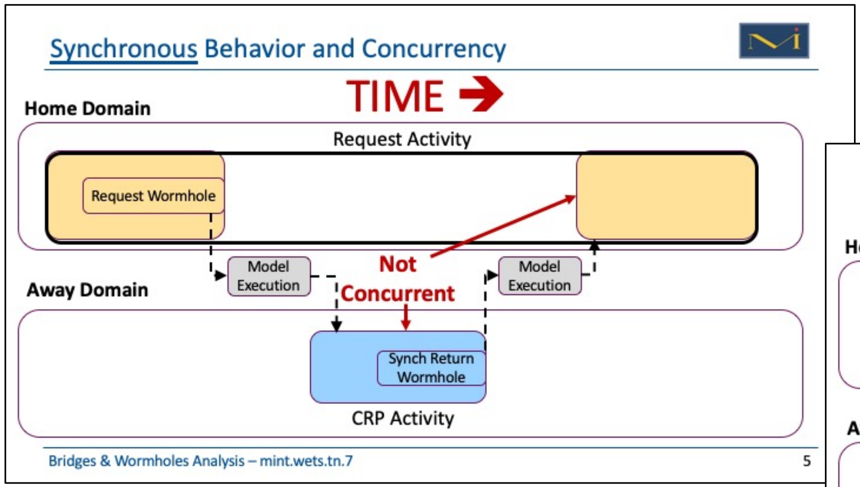
Bridges and Wormholes v2

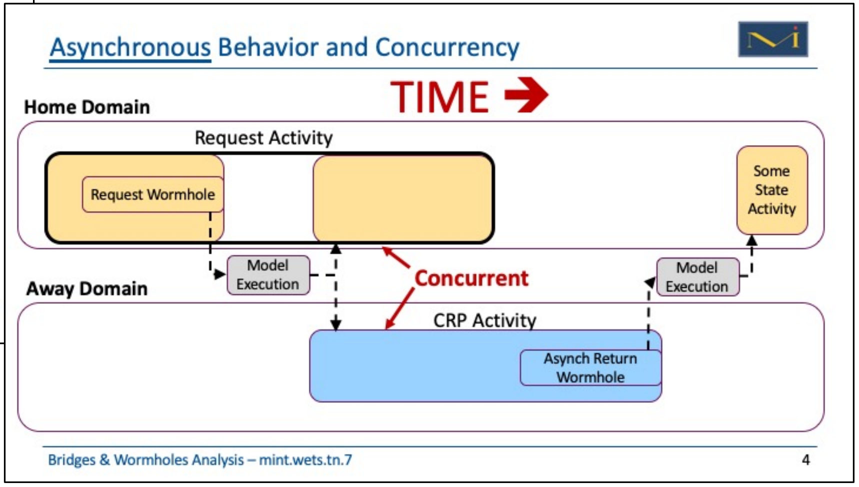
- 2 Request Wormholes, Sync and Async
- New data type, <u>Return Mechanism</u> w/subtypes:
  - Transfer Vector, Return Coordinates, Neither
- 1 Return Wormhole
  - CRP provides stored <u>Return Mechanism</u>
  - MX uses subtype to provide requested behavior

Improvements

- A CRP can provide both sync and async behavior based on subtype of <u>Return Mechanism</u>
- Home does not need to match CRP behavior
- Easy to determine Home's desired behavior from the Request in AL

When We Last Visited… Concurrency





Understanding and modeling concurrency correctly is essential for building models that can successfully execute in multiple software architectures.

Aspects of Concurrency when Modeling a Bridge Request

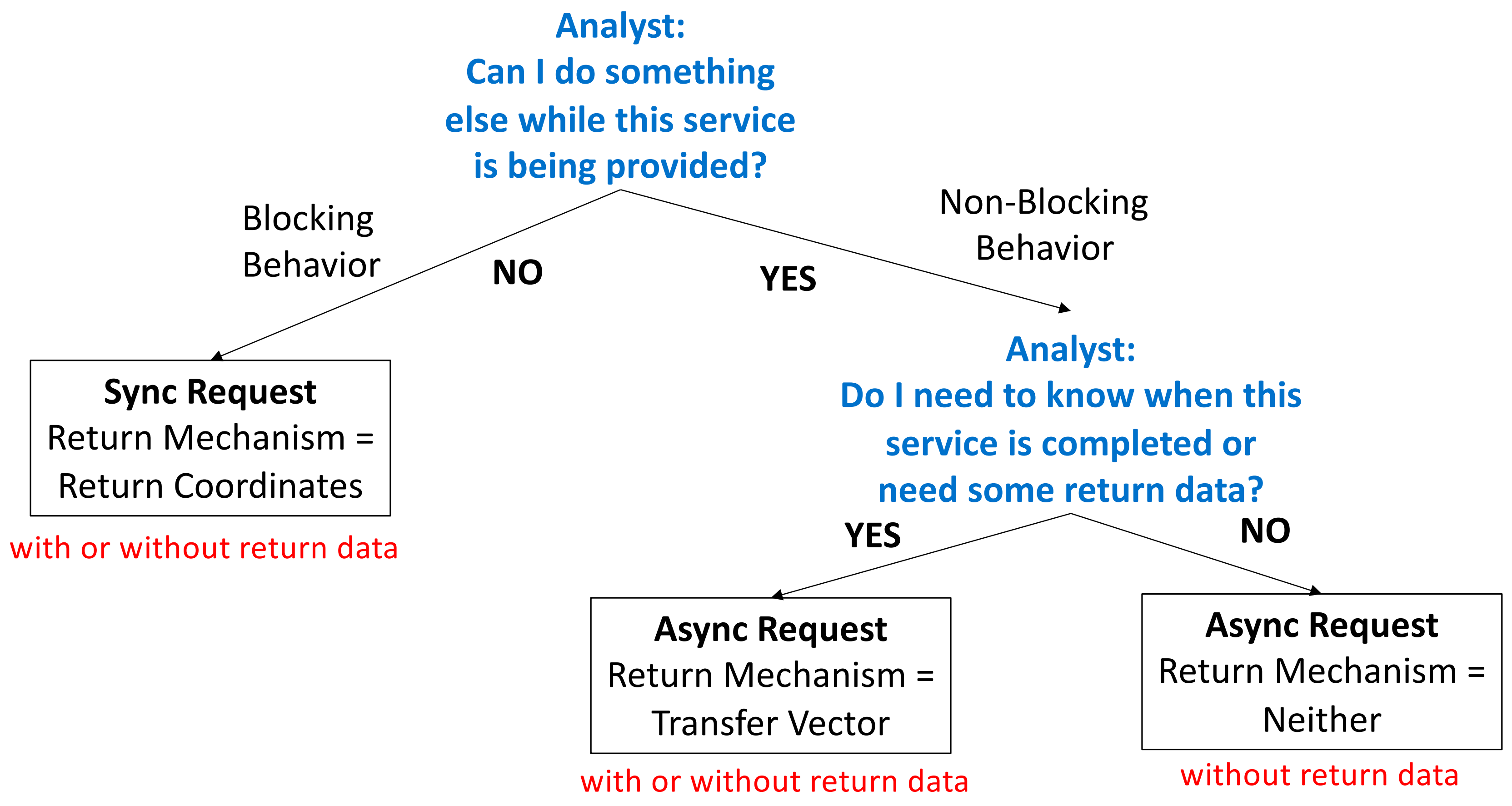
Behavior → Synchronous or Asynchronous for HOME

- A synchronous bridge request <u>blocks</u> the request action in HOME while the service is being provided
- An asynchronous bridge request, <u>does not block</u> the request action in HOME while the service is being provided.

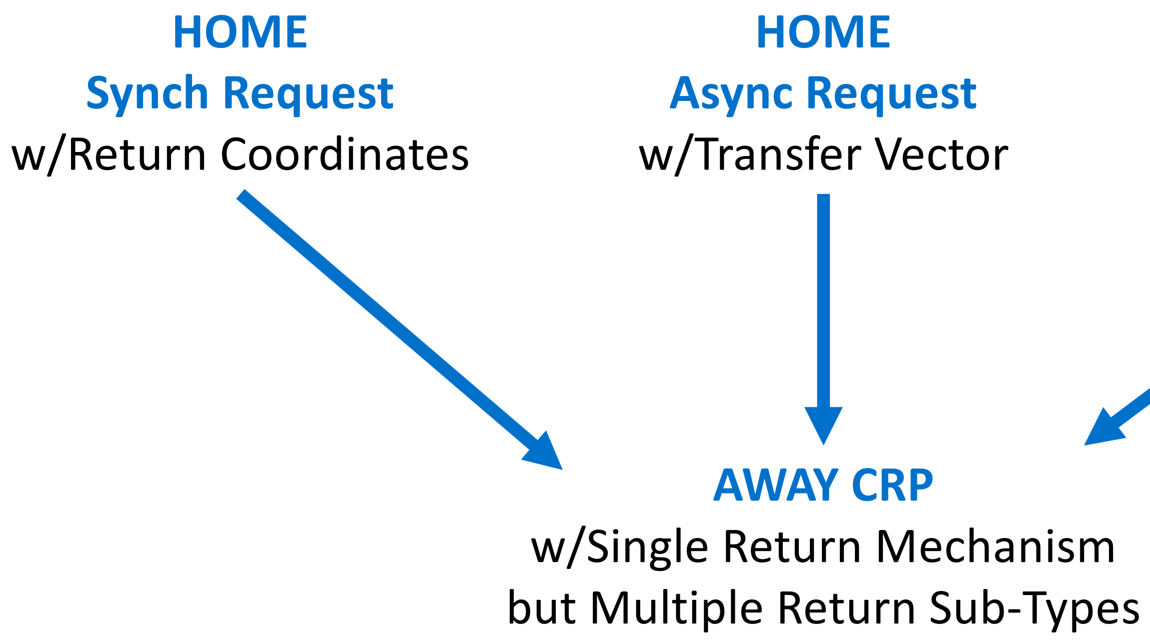
Service Completion Notification → HOME knows requested service is completed

- A synchronous bridge request always gets an implicit service completion notification by virtue of the fact that it gets unblocked when the service is completed.
- An asynchronous bridge request can choose to get a completion notification or not, in the form of a Return Event, depending on whether or not the HOME domain needs one for control or data purposes.

Clarity on Concurrency when Modeling a Bridge Request


Analyst:
Can I do something else while this service is being provided?
Blocking Behavior
NO
YES
Non-Blocking Behavior
Sync Request
Return Mechanism = Return Coordinates
with or without return data
Analyst:
Do I need to know when this service is completed or need some return data?
YES
NO
Async Request
Return Mechanism = Transfer Vector
with or without return data
Async Request
Return Mechanism = Neither
without return data

Lack of Constraints on Matching Behavior w/B&W v2

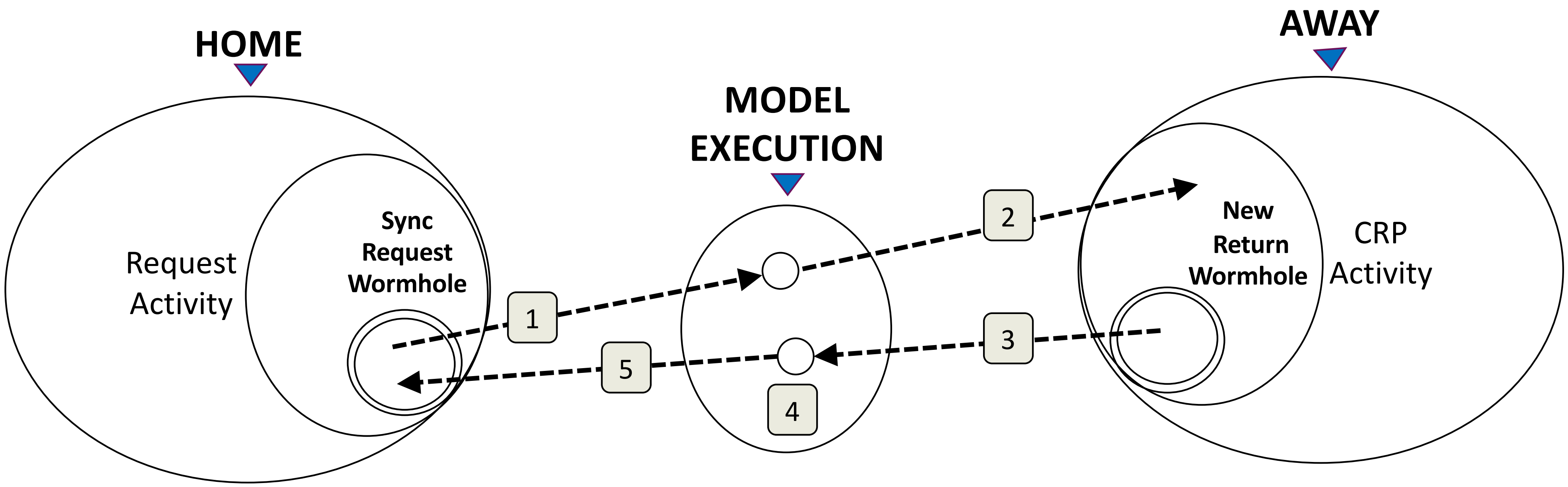


NOTE: I/O Data must also match

Providing Both Synchronous and Asynchronous Behavior

- A CRP is an activity in AWAY that provides a requested service.
- A CRP can be activated by an event or an invocation.
- A CRP will be provided any input data and a Return Mechanism on activation.
- A CRP has some amount of work to do to provide the requested service and may or may not have data to return to HOME.
- When that work is done and any return data is available, the CRP invokes the Return wormhole, providing the Return Mechanism and any return data to the MX.
    - If the Return Mechanism is a Return Coordinate, the data (if any) and control can be returned to the Sync Request action using the provided Return Coordinates
    - If the Return Mechanism is a Transfer Vector, the data (if any) can be returned to the HOME domain via the Return Event created from the Transfer Vector. Control has already been returned to the Async Request action earlier.
    - If the Return Mechanism is Neither, there is nothing to do. Control has already been returned to the Async Request action earlier and HOME doesn't require a notification or data.

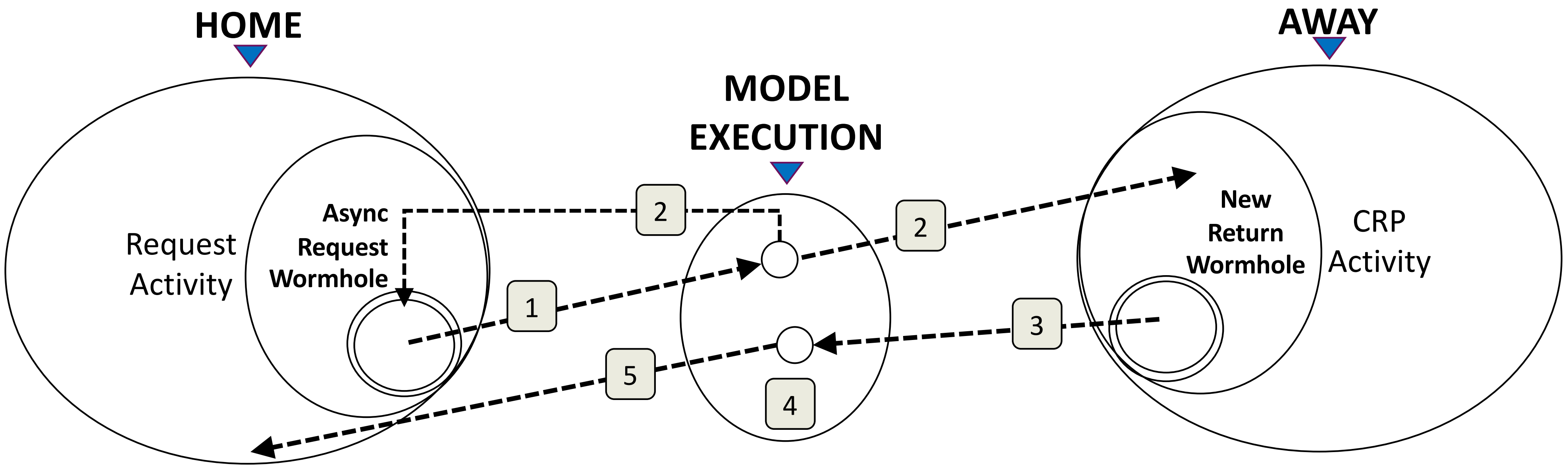
New <u>Synchronous</u> Bridge Scenario



1. A Sync Request Wormhole action executes in **Home** which suspends the Wormhole action and causes MX to…
2. Initiate activation of the associated CRP activity (w/input data + Return Mechanism = Return Coordinates) in **Away**
3. The CRP activity executes until, a New Return Wormhole action executes in the CRP Activity in **Away** , providing a Return Mechanism which causes MX to…
4. Recognize that the Return Mechanism subtype is a Return Coordinates and…
5. Resume the <u>Request Wormhole</u> action in **Home** (w/requested return data, if any).

Red = New

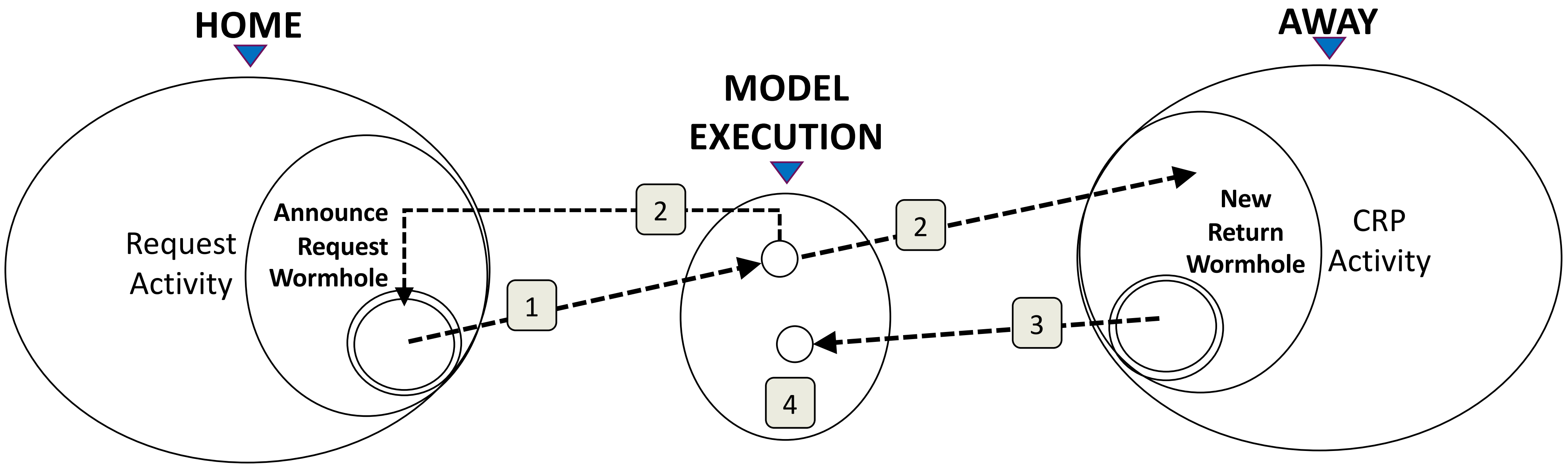
New Asynchronous Bridge Scenario w/Notification



1. An Async Request Wormhole action executes in **Home** which suspends the Wormhole action and causes MX to…
2. Initiate activation of the associated CRP activity (w/input data + Return Mechanism = Transfer Vector) in **Away** and resume the Request Wormhole action in **Home**.
3. The CRP activity executes until a New Return Wormhole action executes in the CRP Activity in **Away** , providing a Return Mechanism which causes MX to…
4. Recognize that the Return Mechanism is a Transfer Vector and…
5. Post the "Populated Return Vector" event in **Home**

Red = New

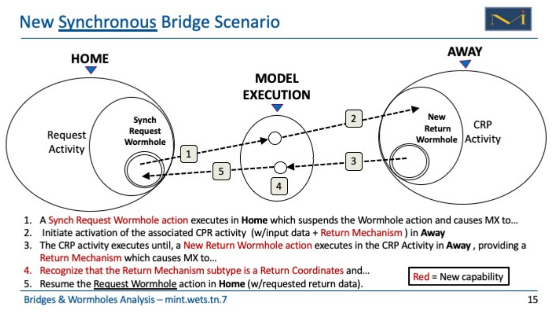
New Asynchronous Bridge Scenario w/o Notification or Data



1. An Announce Request Wormhole action executes in **Home** which suspends the Wormhole action and causes MX to…
2. Initiate activation of the associated CRP activity (w/input data + Return Mechanism = Neither) in **Away** and resume the Request Wormhole action in **Home**.
3. The CRP activity executes. If a New Return Wormhole action is invoked in the CRP Activity in **Away**, it provides a Return Mechanism which causes MX to…
4. Recognize that the Return Mechanism is a Neither and does nothing.

Red = New

Support for New Bridge Scenarios?

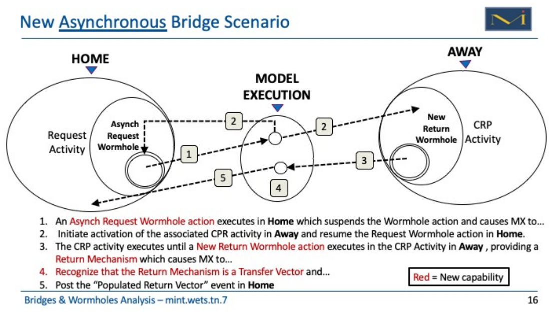


New Synchronous Bridge Scenario

1. A Synch Request Wormhole action executes in **Home** which suspends the Wormhole action and causes MX to…
2. Initiate activation of the associated CPR activity (w/input data + Return Mechanism) in **Away**
3. The CRP activity executes until, a New Return Wormhole action executes in the CRP Activity in **Away** , providing a Return Mechanism which causes MX to…
4. Recognize that the Return Mechanism subtype is a Return Coordinates and…
5. Resume the Request Wormhole action in **Home** (w/requested return data).

Red = New capability

Bridges & Wormholes Analysis – mint.wets.tn.7 15

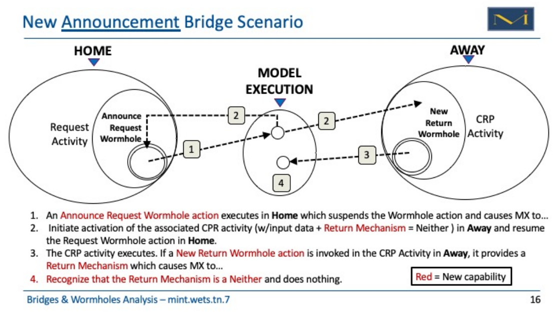


New Asynchronous Bridge Scenario

1. An Asynch Request Wormhole action executes in **Home** which suspends the Wormhole action and causes MX to…
2. Initiate activation of the associated CPR activity in **Away** and resume the Request Wormhole action in **Home**.
3. The CRP activity executes until a New Return Wormhole action executes in the CRP Activity in **Away** , providing a Return Mechanism which causes MX to…
4. Recognize that the Return Mechanism is a Transfer Vector and…
5. Post the "Populated Return Vector" event in **Home**

Red = New capability

Bridges & Wormholes Analysis – mint.wets.tn.7 16



New Announcement Bridge Scenario

1. An Announce Request Wormhole action executes in **Home** which suspends the Wormhole action and causes MX to…
2. Initiate activation of the associated CPR activity (w/input data + Return Mechanism = Neither) in **Away** and resume the Request Wormhole action in **Home**.
3. The CRP activity executes. If a New Return Wormhole action is invoked in the CRP Activity in **Away**, it provides a Return Mechanism which causes MX to…
4. Recognize that the Return Mechanism is a Neither and does nothing.

Red = New capability

Bridges & Wormholes Analysis – mint.wets.tn.7 16

Supported?

As supported as the Scenarios for the B&W v1 Approach

Summary

Results of Analysis of B&W Control

- Focusing on the types of behavior helped clarify
  - matching HOME and AWAY and
  - the constraints behavior placed on the matching.
- Providing step-by-step diagrams of the scenarios helped
  - understand how the behaviors worked and
  - assess the approach's viability.
- Considering the concurrency of execution between Home and Away was more to the point than considering execution to be synchronous or asynchronous.

Possible Benefit of Proposed B&W Improvement

- Removing the constraints behavior places on the matching HOME requests with AWAY services would improve the usefulness of the AWAY services.

But… there is still (much) more work to be done.

Questions? Comments?? Suggestions???

Challenges & Questions w/Bridges & Wormholes Approach

- Need to handle a request wanting different return events based on results of service invocation (e.g., Fail or Succeed) PLUS can this map to a synchronous behavior?
- Generic, table-based bridge code is allowed by B&W. can this handle all the "semantic shift" work necessary?
- Can you have a mismatch between Home & Away wrt the number of return parameters as long as Away provides more that Home wants?
- Why model a CRP as an External Event or a Synchronous Service?:
  - A CRP is modeled as an External Event if the service is dependent on the state of of the instance to which the event is sent. This will initiate a state activity.
  - A CRP is modeled as a Synchronous Service if the service is not dependent on any state machine's state. This will initiate a domain level operation.
- CONSIDER: An AWAY service activated by an event can initiate different state activities depending on the state of the instance state machine when the EE event arrives.