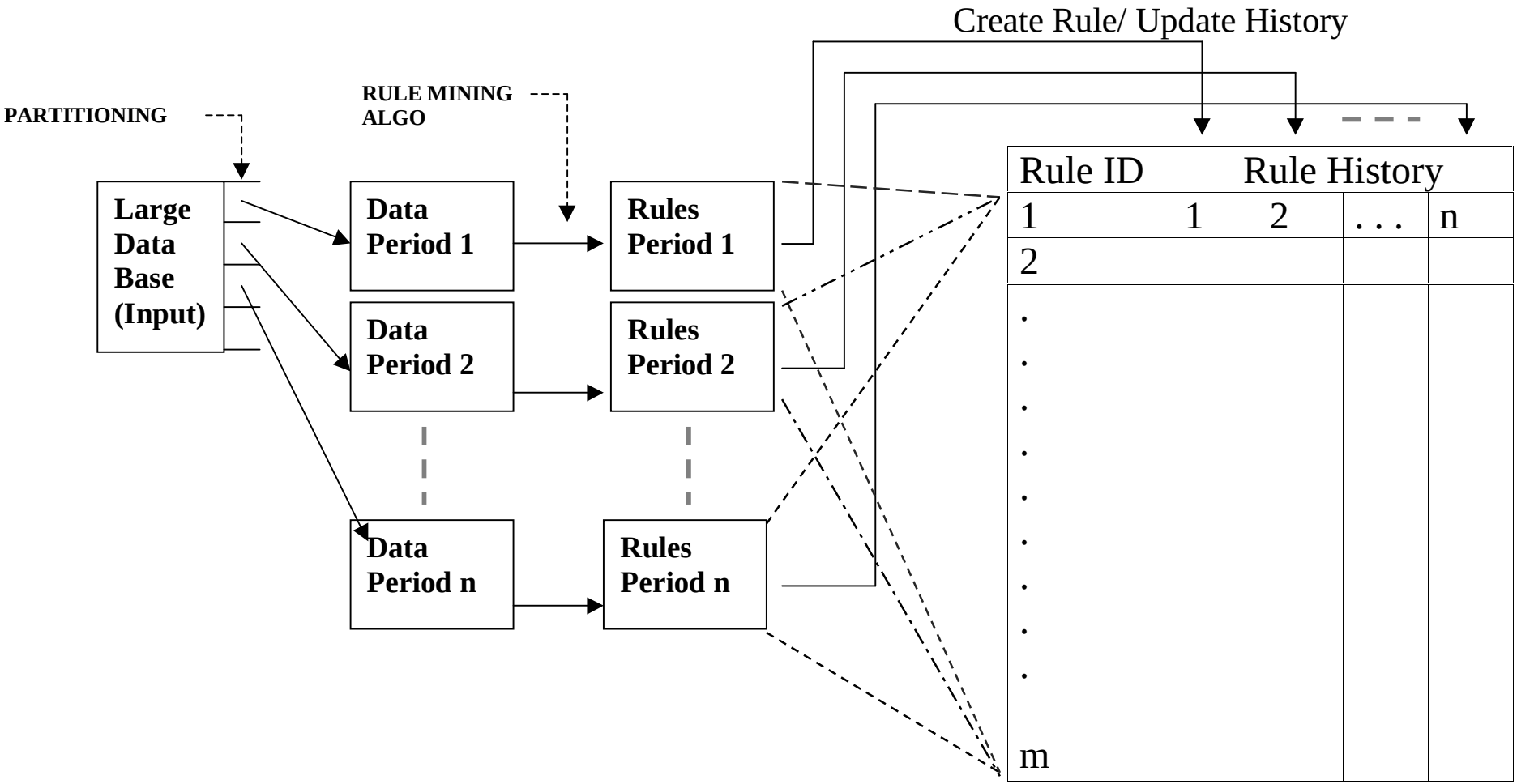
Why Active Data Mining?

- Huge amount of Data, too many rules discovered.
- Frequent Database Update.
- Online Continuous Processing needed for the updated data.
- History based analysis: good for forecasting trends.
- Shape Queries – more powerful language than regular expression based languages. Very good for extracting trend specific rules.
- Triggers: Incremental and Automatic Forecasting.

- Active Data Mining Process

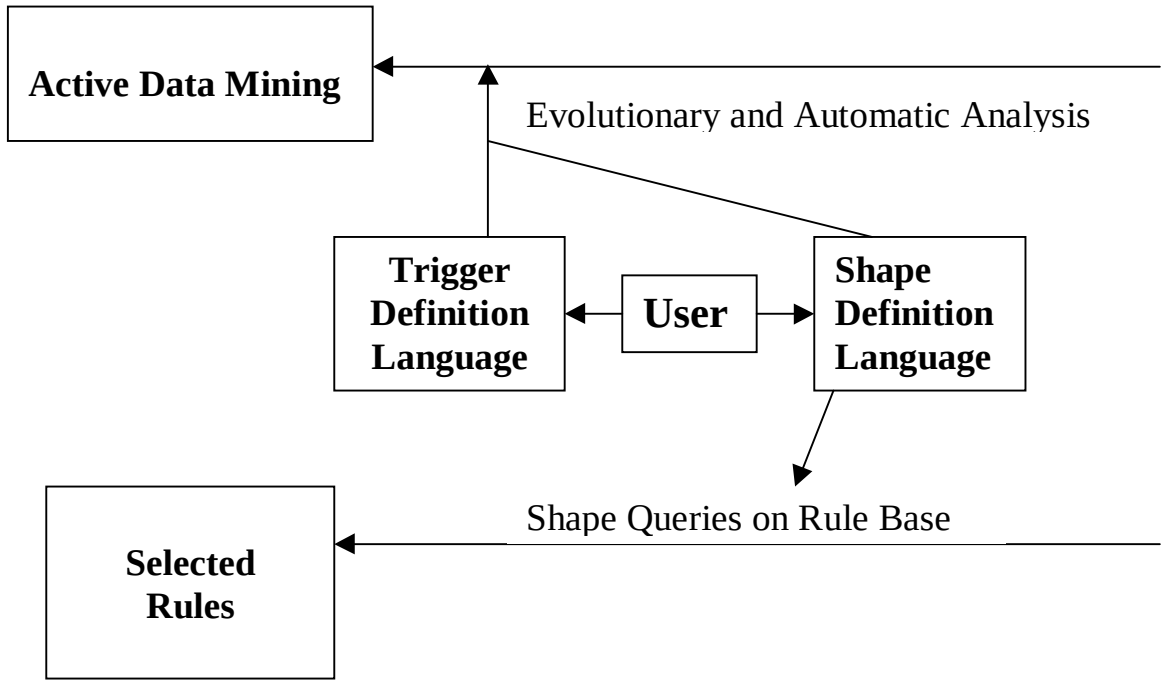
(Part 1: Generation of Rule Base)



Rule Base (Output)

Active Data Mining Process

(Part 2: Querying the Rule Base)



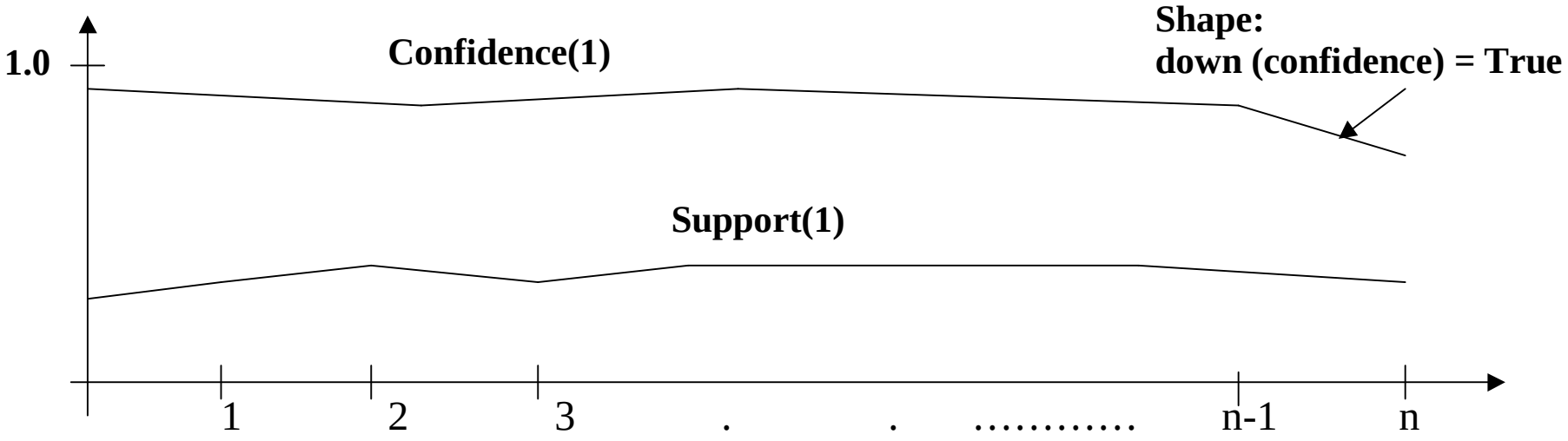
Rule ID	Rule History			
1	1	2	. . .	n
2				
.				
.				
.				
.				
.				
.				
.				
.				
m				

Rule Base
(Constantly updated with New Data)

An Example of a Rule Parameter History

History of Parameters Support & Confidence History for Rule X→Y with Rule ID 1

Rule ID	Rule History			
	Period 1	……..	Period n-1	Period n
1 (X→Y)	Support : 0.30 Confidence : 0.89	…….	Support : 0.31 Confidence : 0.90	Support : 0.30 Confidence : 0.65



Shape Definition

- Used for defining Queries and Trigger, define by users.
- Typical **elementary shapes**: Up , up , Down , down , disappears, appears, stable , zero etc.

(shape down (param) (change of param is between 0.05 and 0.29))

In the last slide , down(confidence(1))= True for n^{th} transition, why ?

- **Complex Shapes**: Shapes defined using simpler shapes : Spike, DoublePeak, bullish, drift etc.

```
(shape spike (upcnt dncnt)
    (concat (atleast upcnt (any up Up))
            (atleast dncnt
                    ( any down Down))))
```

Defining Shape Queries using Shapes

- Used to retrieve rules satisfying a shape criteria
- **Syntax -**

```
(query (shape history-spec))
```

- **Syntax of history-spec –**

```
history-name start-time end-time
```

where time is between 1 & n

- **Global search** (matching over complete history 1 to n) –

```
history-name start end
```

- **Return value of Queries –**

```
((query1, <time1-time2>) (query2), <time3-time4>) …… )
```

- **An Example** : (query down(confidence) start end)
  will return **(1, <n-1 to n>)** as one of the conforming query.

Triggers

- Triggers activate because of **Events**.
- **Typical Events** defined by user : CreateRule, Updatehistory.
  - CreateRule : Occurs when a new rule is created in Rule-Base
  - UpdateHistory: Occurs when existing rules get updated.

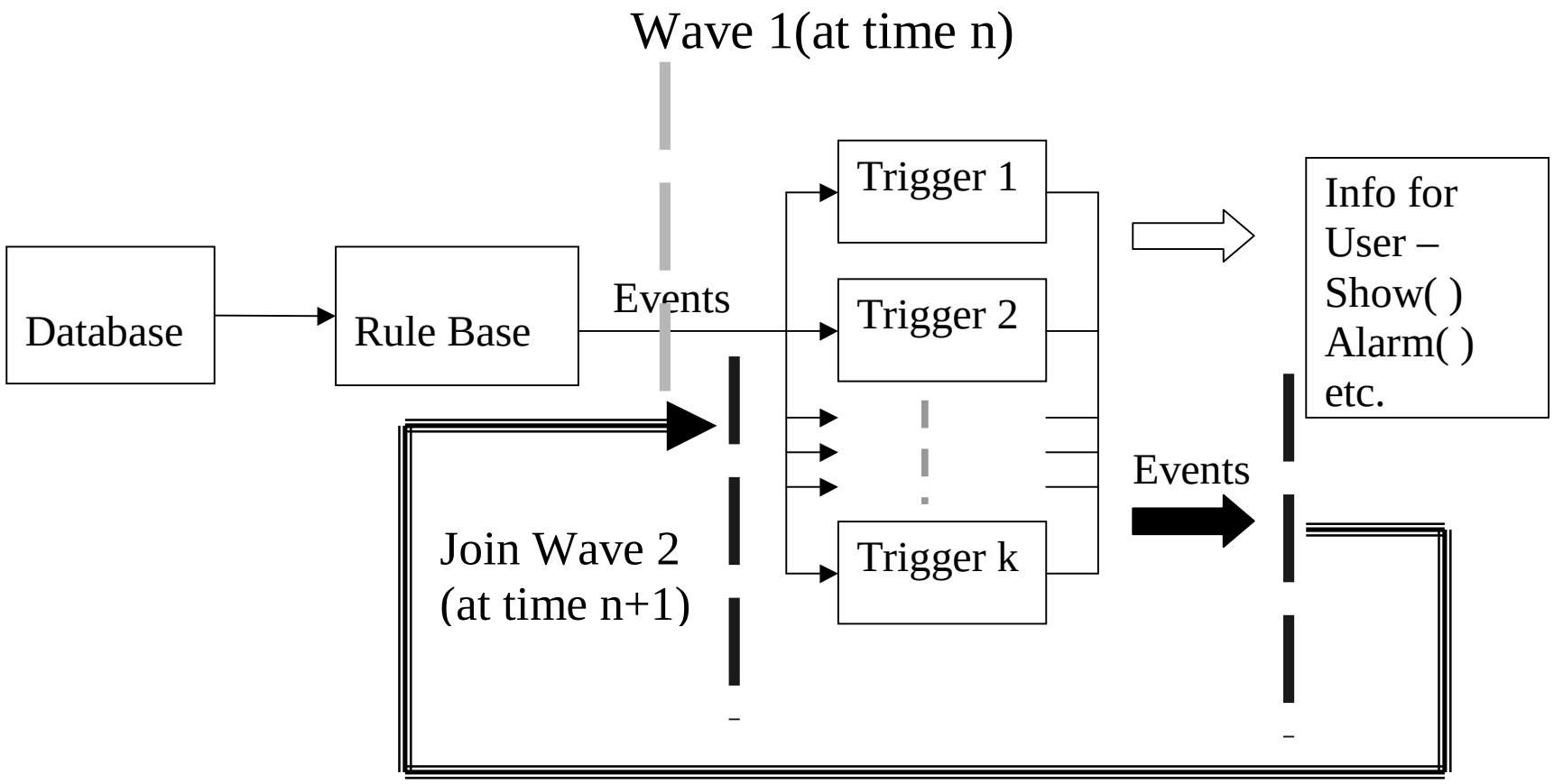
- **Syntax –**

(trigger *trigger-name*
(events *event-spec*)
(conditions (*shape history-spec*))
(actions action-spec))

- **Actions of a Trigger –**

- Action-spec in a trigger is a list of user defined functions like **action** or **show** which is executed.
- Cascaded Triggers : Triggers with action which as an event triggering another Trigger.

Wave Execution Model of a Trigger



Conclusions

- Implementation on AIX platform as part of project Quest at IBM
- Tested against two large 5-year period data-sets of 2.9 and 6.8 million transactions of a mail-order company.
- Data division on a monthly basis i.e. time intervals n= 60.
- Parameters support, confidence and (support X confidence) used.
- Seems very attractive for large-scale commercial implementation.

- Improvements suggested :

Recompute Triggers only on incremented history and not the full history as is being done now to improve performance.

- My review : Paper very comprehensive and well written. Interesting concepts.
- No performance comparisons or benchmarking included.