DECISION TABLES: A PRIMER

How to Use TableSpeak™

Version 2.0

By Ronald G. Ross

Business Rule Solutions, LLC
The Business Rule Technique Company

www.brsolutions.com

ACKNOWLEDGEMENTS

Author: Ronald G. Ross
Co-Founder & Principal, Business Rules Solutions, LLC
Executive Editor, www.BRCommunity.com
Chair, Business Rules & Decisions Forum Conference
Twitter: Ronald_G_Ross

Editor: Keri Anderson Healy
Editor, www.BRCommunity.com

The author would like to acknowledge …

- **Gladys S.W. Lam**, Co-Founder & Principal of Business Rule Solutions, LLC, who co-developed the concepts and approach described in this Primer.

Colleagues who most graciously reviewed this work and offered substantial improvements include …

- **Both Releases:** Kristen Seer, Jan Vanthienen
- **First Release:** Don Baisley, Rik Gerrits, John Healy, Keri Anderson Healy, Silvie Spreeuwenberg

EXECUTIVE SUMMARY

Many decision tables are too technical. They are aimed at software developers, not business people. Do they have to be that way?

No. Our approach, **TableSpeak**, stays as close as possible to formats intuitive and natural for business people. TableSpeak does not require business people or business analysts to:

- Learn any special new format and related conventions.
- Use a single format even if not optimal for a given problem.

TableSpeak is:

- Aimed at supporting ongoing coordination of **operational business decisions**.
- A natural follow-on to **DecisionSpeak**[1], our top-down approach for business-oriented decision analysis.
- Like DecisionSpeak, a part of the comprehensive BRS methodology, **IPSpeak®**.

TableSpeak emphasizes:

- Deliberate structure, to ensure business-friendly representation of **decision logic** and on-target implementation.
- Careful communication of meaning, to avoid ambiguity and misinterpretation.
- Protections for the integrity (correctness) of content, to avoid faulty decisions.
- **Declarative** representation, to ensure decision logic remains platform-independent and highly re-usable.
- **Single-sourcing**, to ensure that any given specification will be as easy as possible to find and to change quickly and reliably.

Refer to Appendix 1 for a quick overview of TableSpeak.

Decision tables are by no means a perfect form of representation, nor can they be used for a great many **business rules**. They are *not* a silver bullet. Be aware that:

- Some styles of decision table design prove better than others.
- You will likely need automated support for thorough analysis of larger decision tables.

Used correctly, however, decision tables can be extremely useful in representing, analyzing, implementing, and managing certain forms of **know-how**. They fit handily with many implementation platforms, including decision management systems. They provide significant opportunities for ROI.

We believe decision tables are a key technique for business analysis. This discussion introduces you to everything you need to know about understanding and designing decision tables.

[1] *Decision Analysis: A Primer – How to Use DecisionSpeak™ and Question Charts (Q-Charts™):* http://www.brsolutions.com/IPSpeakPrimers

TABLE OF CONTENTS

FUNDAMENTALS

PART 1: THE BASICS OF DECISION TABLES

1.1 WHAT A DECISION TABLE IS

A **decision** is *a determination requiring* **know-how** *or expertise; the resolving of a question by identifying some correct or optimal choice*.

A decision should represent the best or most appropriate answer (outcome) among potential outcomes for some weighty (non-trivial) choice the business must make repetitively in day-to-day business activity.

Decisions appropriate for **decision analysis** are always **operational business decisions** – *not* programming, personal, strategic, or governance decisions. To emphasize the point, this Primer always says *operational business decision*, not simply *decision*.[2]

Decision tables are an excellent means to represent the decision rules on which an operational business decision is based.

Be fully aware, however, that decision tables cannot be used to capture a great many business rules.[3] Don't be misled in that regard.

1.1.1 The Question a Decision Table Addresses

An example of a simple decision table is presented in Figure P1-1. This decision table addresses the question *What coat should be worn?.*

Figure P1-1. Example of a Simple Decision Table

		Is it rainy?	
		Yes, it's rainy	***No, it's not rainy***
Is it cold?	***Yes, it's cold***	*lined raincoat*	*wool overcoat*
	No, it's not cold	*unlined raincoat*	*none*

The decision table in Figure P1.1 can be used to answer four specific variations of the question *What coat should be worn?*, as follows.

1. What coat should be worn *if it is cold and rainy?*
2. What coat should be worn *if it is cold and not rainy?*
3. What coat should be worn *if it is not cold and not rainy?*
4. What coat should be worn *if it is not cold but rainy?*

The answers can be found in the appropriate intersection cells (for convenience, lightly colored), starting at the top row on the left, then reading clockwise.

[2] There is much confusion in the field on this point. The practitioner is urged to carefully review the important target-problem clarifications presented in Appendix 2 of the companion Primer on **DecisionSpeak**.
http://www.brsolutions.com/IPSpeakPrimers

[3] Refer to Appendix 3 of the companion Primer on DecisionSpeak regarding the distinction between decision rules and other important kinds of **business rules** that need to be captured and managed.
http://www.brsolutions.com/IPSpeakPrimers

1.1.2 Decision Rules and Outcomes

The answers to the four specific questions above represent four decision rules, which could be expressed as follows.

1. A *lined raincoat* should be worn if it is cold and rainy.
2. A *wool overcoat* should be worn if it is cold and not rainy.
3. A *coat need not* be worn if it is not cold and not rainy.
4. An *unlined raincoat* should be worn if it is not cold but rainy.

A **decision rule** is a **business rule** *that guides the making of an operational business decision, a business rule that provides a specific answer to a selective question*.

A significant benefit of using decision tables is that there is no need to write out the decision rules as above (unless desired for clarification). Appropriate outcomes simply appear in the decision cells of the decision table.

An **outcome** is the *result, conclusion or answer given by a decision rule to a selective question being asked.* Example:

> *Lined raincoat* is the outcome given by the decision rule *A lined raincoat should be worn if it is cold and rainy.*

The outcome given by a decision rule is selected from among a set of **potential outcomes**, all the individual outcomes permitted for answering the overall question. Potential outcomes for the decision table in Figure P1-1 include at least:

1. lined raincoat
2. wool overcoat
3. no coat (none)
4. unlined raincoat

Other potential outcomes might exist besides these four. Identifying the complete set of potential outcomes is always an important concern in decision analysis.

Incidentally, did you notice that *none* is not a coat!? For discussion of that interesting word refer to Appendix 7.

1.1.3 What a Decision Table Is

What is a decision table?

A **decision table** is simply *a structured means of visualizing decision rules in rows and columns*. A decision table in **TableSpeak** *always*:

- Addresses questions – once generally and many times specifically.
- Implicitly represents decision rules.
- Provides answers (outcomes) selected from among potential outcomes.

A key word in the definition of decision table is *visualizing*. Decision tables are a means of representing decision rules for viewing and updating in the best (most business-friendly) way possible.

Decision tables are *not* a data management or database scheme – an entirely different issue. Some practitioners – and experts as well – are quite confused on this important point.[4]

Not every table, of course, is a *decision* table. Tables can be used productively for a great many purposes. If a table does not represent

[4] In terms of three-schema architecture, decision tables are an *external* schema. Refer to: http://en.wikipedia.org/wiki/Three_schema_approach

decision rules *fully* the table is *not* a decision table.

Refer to Appendix 2 for examples of other kinds of tables potentially useful in analyzing or communicating decision logic, including **look-up tables** and **scenario tables**.

1.2 THE CASES THAT A DECISION TABLE ADDRESSES

What we want from a **decision table** are the answers to a question. First, however, the decision table must be *structured* properly to provide and manage these answers in optimal fashion.

The most fundamental idea in structuring a decision table is that it addresses particular *cases* of interest.

A **case** is simply *some particular situation* – nothing more, nothing less. Cases might be called *scenarios*, but **TableSpeak** prefers the term *case* to avoid any sense of events or actions – i.e., 'flow'. Think of a case as a snapshot of circumstances that at least momentarily don't flow.

The decision table in Figure P1-1 specifically addresses the following four cases:

- *It is cold and rainy.*
- *It is cold but not rainy.*
- *It is not cold and not rainy.*
- *It is not cold but rainy.*

These four cases are composite. Each is clearly based on two factors rather than just one:

1. *Is it cold?*
2. *Is it rainy?*

In **TableSpeak** these factors are called *considerations.*

A **consideration** is *a factor in making an* **operational business decision**; *something that can be resolved to two or more* **cases.**

Important points:

How should considerations be worded?

> Although considerations can always be worded as questions (as above) TableSpeak does not insist on that. For example, the two considerations above could be called *temperature* and *precipitation*, respectively.
>
> The key is to word or name each consideration in a clear, business-friendly fashion.

How many considerations should a decision table include?

> The decision table in Figure P1-1 involves two considerations.
>
> Many decision tables, of course, involve more than that. As more considerations are added, the complexity of representation, analysis and management naturally escalates.
>
> *It is generally recommended that the number of considerations for a decision table not exceed 7.*
>
> This threshold, discussed in Part 2, is called the **complexity threshold**. DecisionSpeak and TableSpeak offer pragmatic techniques to ensure this important guideline can be followed.

What kinds of cases can considerations produce?

Considerations produce two fundamental kinds of cases, *elemental* and *intersection*, as discussed below.

1.2.1 Elemental Cases

An **elemental cases** is a case produced directly from a single consideration. Examples:

1. The consideration *Is it cold?* produces the two elemental cases:

- *Yes, it's cold.*
- *No, it's not cold.*

2. The consideration *Is it rainy?* also produces two elemental cases:

- *Yes, it's rainy.*
- *No, it's not rainy.*

How should elemental cases be worded?

Elemental cases need not be specified in quite so wordy a fashion as above. For example, simply *yes* and *no* would probably suffice. TableSpeak, however, always focuses on avoiding *any* possibility of ambiguity or misinterpretation. Good judgment in this regard should be exercised.

Note that cases are never worded as questions.

How many elemental cases can a consideration produce?

The two considerations above are binary – they each produce two elemental cases. Many considerations produce more than two cases. Examples:

- *category of customer* might produce three cases – *platinum, gold* and *silver*.
- *province of Canada* produces ten cases.
- *zip code* or *postal code* can produce many thousands of elemental cases.

Each consideration should always produce *at least two* elemental cases.

A factor that results in only one elemental case (in scope) – e.g. *California* – can be handled in some better way. In TableSpeak such a consideration might be treated as a **scope item** or an **exception**. For discussion refer to Part 1.5.2 and Part 3.2, respectively.

What quality considerations apply to the elemental considerations?

It is extremely important that the set of elemental cases for a consideration be:

1. **Exhaustive** (inclusive of all **cases in scope**).
2. **Disjoint** (non-overlapping).

Refer to Part 5.2 for discussion of these quality criteria and for illustrations of **anomalies** that can occur in decision tables when the criteria are not satisfied.

1.2.2 Intersection Cases

An **intersection case** is a compound case involving two or more elemental cases. Example:

The intersection case *It is cold and rainy.* is produced from the two considerations:

- *Is it cold?*
- *Is it rainy?*

A decision table with only one consideration does not comprise any intersection cases. Since most decision tables involve two or more

considerations, however, most do involve intersection cases.

How intersection cases are handled is a central issue in structuring decision tables. The next section explains.

1.3 STRUCTURAL STYLES FOR DECISION TABLES

The basic structure of all **decision tables** is based on rows and columns that intersect in a physical matrix. If there are *m* rows and *n* columns, the matrix can contain *m* × *n* cells.

No cell means *anything* until you indicate what it means. All cells are empty until you populate them. The physical matrix is simply a blank canvas.

TableSpeak makes two fundamental (and common-sense) assumptions about how this blank canvas can be used to represent **decision rules**.

Assumption 1. Each cell should always (a) represent just one kind of thing, and (b) hold just one kind of thing. If not, there's no way to guarantee uniform meaning in use of the physical matrix. All bets are off in trying to interpret and implement the result.

Assumption 2. All parts of each decision rule should be represented such that they are easily understood *as a unit*. Decision rules are strings of meaning, like sentences. The strings can't be represented randomly; the parts have to come together or the 'sentence' meaning is lost.

The most basic question in formatting decision tables is therefore how many kinds of things can decision rules involve?

The answer is *three* kinds of things:

1. One or more **consideration(s)** and their **elemental cases**.
2. **Intersection cases** (assuming more than one consideration).
3. **Outcomes**.

How can a physical matrix be used to represent these three kinds of things as easily understood units (decision rules)?

Fortunately, the options in that regard are quite limited. In fact there are *just three* basic styles in employing a physical matrix to create decision tables. Two of the styles are simply variations on the same theme.

Each style is introduced below, then discussed in greater detail in Part 2.

1.3.1 Intersection Style

The simplest format for decision tables, the one friendliest for business people, is the **intersection style** (sometimes called *crosstab* style).

Figure P1-1 was organized using this style. As that example illustrated, **considerations** are organized using *each* of the two dimensions (rows and columns). All **elemental cases** for each consideration are *literally* 'intersected' using rows and columns.

The cells of the leftmost column and of the topmost row represent elemental cases. Cells where these rows and columns meet represent **intersection cases**. To illustrate, all the cases represented by the *What coat should be worn?* decision table in Figure P1-1 are labeled in Figure P1-2.

Figure P1-2. How Cases Are Represented in the *What coat should be worn?* Intersection-Style Decision Table

Is it rainy?

Is it cold?	*elemental case*	*elemental case*
elemental case	*Intersection case*	*Intersection case*
elemental case	*Intersection case*	*Intersection case*

Although the physical cells where rows and columns meet in Figure P1-2 *represent* intersection cases, those cells *hold* outcomes (and can therefore be called **decision cells**). Figure P1-3 illustrates.

Figure P1-3. Outcomes in the *What coat should be worn?* Intersection-Style Decision Table

consideration b

consideration a	*elemental case b1*	*elemental case b2*
elemental case a1	*outcome for a1/b1*	*outcome for a1/b2*
elemental case a2	*outcome for a2/b1*	*outcome for a2/b2*

The intersection style is physically notable in two basic regards:

1. No extra cells are required to hold outcomes. The cells representing intersection cases do double duty.
2. No elemental cases are ever repeated anywhere in the decision table.

1.3.2 One-Rule-Per-Row Style

A second format for decision tables is the **one-rule-per-row style**.

In this style each row in its entirety (below the topmost) represents a **decision rule**. Figure P1-4 recasts the *What coat should be worn?* decision table in Figure P1-1 into this style.

This format features:

- One column per **consideration**, starting on the left. The consideration is labeled in the top row of the column, then **elemental cases** appear in the cells below it.
- An additional column on the right provides **decision cells**; that is, physical cells to hold the appropriate **outcome** for each row (decision rule).

The additional column could have just as easily appeared on the left of the physical matrix. Since most Western languages are written from left to right, however, the right-hand side generally proves more natural for speakers of those languages.

Placement on the right unfortunately does nothing to discourage practitioners from using the style to express or interpret the content of

Figure P1-4. The *What coat should be worn?* Decision Table in the One-Rule-Per-Row Format

Is it cold?	Is it rainy?	*What coat should be worn?*
Yes, it's cold	**Yes, it's rainy**	*lined raincoat*
Yes, it's cold	**No, it's not rainy**	*wool overcoat*
No, it's not cold	**Yes, it's rainy**	*unlined raincoat*
No, it's not cold	**No, it's not rainy**	*none*

the rows as *If … and if … then do this*. Such use represents a **procedural** rather than a **declarative** approach. TableSpeak recommends never using a decision table for that purpose. Refer to Part 2.6 for additional discussion.

Figure P1-5 explicitly identifies the elemental cases and outcomes evident in Figure P1-4.

bit more approachable. Note that repetition of elemental cases in the second column remains.

In the one-rule-per-row style there is no literal intersection of rows and columns to produce **intersection cases**. Rather, elemental cases are effectively *concatenated* (united in a series) to form intersection cases. Figure P1-7 illustrates.

Figure P1-5. Elemental Cases and Outcomes Identified for the *What coat should be worn?* Decision Table in the One-Rule-Per-Row Style

Is it cold?	Is it rainy?	*What coat should be worn?*
elemental case a1	***elemental case b1***	*outcome for intersection case a1b1*
elemental case a1	***elemental case b2***	*outcome for intersection case a1b2*
elemental case a2	***elemental case b1***	*outcome for intersection case a2b1*
elemental case a2	***elemental case b2***	*outcome for intersection case a2b2*

To simplify the representation of a one-rule-per-row decision table, elemental cases in consecutive rows that are alike are often consolidated. Figure P1-6 illustrates such consolidation for Figure P1-4.

Starting with that Figure, elemental cases are also simplified from this point forward in the discussion to just *yes* and *no* to avoid wordiness.

Figure P1-6. The *What coat should be worn?* Decision Table with Consolidation of Elemental Cases in the Leftmost Column

Is it cold?	Is it rainy?	*What coat should be worn?*
yes	**yes**	*lined raincoat*
	no	*wool overcoat*
no	**yes**	*unlined raincoat*
	no	*none*

Now this revised version of the decision table avoids repetition of elemental cases in the leftmost column, rendering the decision table a

Figure P1-7. Intersection Cases Addressed by the *What coat should be worn?* Decision Table in the One-Rule-Per-Row Style

consideration a	consideration b	outcome
intersection case a1b1		*outcome for intersection case a1b1*
intersection case a1b2		*outcome for intersection case a1b2*
intersection case a2b1		*outcome for intersection case a2b1*
intersection case a2b2		*outcome for intersection case a2b2*

Compared to the intersection style, the one-rule-per-row style:

1. Employs an additional column to hold outcomes.

2. Repeats elemental cases for each individual consideration to produce intersection cases.

1.3.3 One-Rule-Per-Column Style

A third format for decision tables is the **one-rule-per-column style**. In this style each column in its entirety (except the leftmost) represents a **decision rule**. Figure P1-8 recasts the *What coat should be worn?* decision table in Figures P1-1 and P1-4 into this style.

Figure P1-8. The *What coat should be worn?* Decision Table in the One-Rule-Per-Column Format

Is it cold?	**yes**	**yes**	**no**	**no**
Is it rainy?	**yes**	**no**	**yes**	**no**
What coat should be worn?	*lined raincoat*	*wool overcoat*	*unlined raincoat*	*none*

This format features:

- One row per **consideration**, starting at the top. The consideration is labeled in the leftmost column of the row, then **elemental cases** appear in the cells to the right of it. (Like elemental cases in the topmost row could have been consolidated.)
- An additional row at the bottom provides **decision cells**; that is, physical cells to hold the appropriate **outcome** for each column (decision rule).

The additional row could have just as easily appeared at the top of the physical matrix. Figure P1-9 illustrates.

TableSpeak recommends (but does not insist upon) this top-row-for-outcomes arrangement for two reasons:

- It most closely aligns with the subject-orientation of well-written business rules. For example, the first decision rule (in the second physical column) would be best written: *A lined raincoat should be worn if it is cold and rainy*.
- It physically emphasizes that decision rules are declarative, rather than procedural. Putting the outcome at the bottom makes it tempting to express or interpret a decision rule as *If … and if … then do this*. As before, TableSpeak recommends never using a decision table for that purpose.

Refer to Appendix 3 for an additional example and a caution in using the top-row-for-outcomes arrangement.

Figure P1-9. The *What coat should be worn?* Decision Table in the One-Rule-Per-Column Format with Outcomes Along the Top Row

What coat should be worn?	*lined raincoat*	*wool overcoat*	*unlined raincoat*	*none*
Is it cold?	**yes**	**yes**	**no**	**no**
Is it rainy?	**yes**	**no**	**yes**	**no**

Figure P1-10. Elemental Cases and Outcomes Identified for the *What coat should be worn?* Decision Table in the One-Rule-Per-Column Style

What coat should be worn?	*outcome for intersection case a1b1*	*outcome for intersection case a1b2*	*outcome for intersection case a2b1*	*outcome for intersection case a2b2*
Is it cold?	*elemental case a1*	*elemental case a1*	*elemental case a2*	*elemental case a2*
Is it rainy?	*elemental case b1*	*elemental case b2*	*elemental case b1*	*elemental case b2*

Figure P1-10 explicitly identifies the elemental cases and outcomes evident in Figure P1-9.

In this style there is no literal intersection of rows and columns to produce **intersection cases**. Rather, elemental cases in effect are effectively *concatenated* (united in a series) to form intersection cases, as illustrated by Figure P1-11.

Compared to the intersection style, the one-rule-per-column style:

1. Employs an additional row to hold outcomes (at top or bottom).
2. Repeats elemental cases for each individual consideration to produce intersection cases.

1.3.4 Row-or-Column-Style

The one-rule-per-column style is very much like the one-rule-per-row style on the two points above.

The difference between the two styles, more perception than substance, is simply that the former uses rows, rather than columns, to hold decision rules and outcomes. Therefore TableSpeak often refers to these two styles jointly as the **row-or-column-style**.

Both styles are less friendly to business people than the intersection style, and as discussed momentarily, are more prone to certain kinds of **anomalies**. Refer to Appendix 8 for additional background on the row-or-column-style.

1.4 BASIC ADVANTAGES OF INTERSECTION-STYLE DECISION TABLES

Besides being very approachable for business people, **intersection-style decision tables** have additional advantages, ones so basic that at first they might not be obvious.

Assume that the **elemental cases** of each **consideration** of an **intersection-style decision table** are listed exhaustively and non-

Figure P1-11. Intersection Cases Addressed by the *What coat should be worn?* Decision Table in the One-Rule-Per-Column Style

outcome	*outcome for intersection case a1b1*	*outcome for intersection case a1b2*	*outcome for intersection case a2b1*	*outcome for intersection case a2b2*
consideration a	*intersection case a1b1*	*intersection case a1b2*	*intersection case a2b1*	*intersection case a2b2*
consideration b				

redundantly – that is, each elemental case appears one and only one time. (Software might be needed to ensure this, but it's a relatively simple matter.)

Then it is *physically* impossible for any **intersection case**:

- Not to be represented at all (i.e., missing).
- To be represented more than once.

Why are these characteristics so important? The former helps ensure **completeness**. The latter provides a single point of change (**single-sourcing**) for each **decision rule**.

To illustrate these advantages, Figure P1-12 includes revisions of the *What coat should be worn?* decision table in Figure P1-1.

Figure P1-12. Empty Decision Cell and Changed Outcome in an Intersection-Style Decision Table

		Is it rainy?	
		yes	***no***
Is it cold?	***yes***	*lined raincoat*	
	no	~~*unlined raincoat*~~ *umbrella*	*none*

Completeness. The upper-right decision cell in this version of the decision table is empty. The omission is highly visible.

In the intersection-style format an empty decision cell signals clearly that an **outcome** – more precisely a decision rule – is missing. In other words, such an **omission** signals the **decision logic** is not complete.

Important:

> *An empty decision cell is always problematic in decision tables.*

No matter what style is used, a decision table should never leave any doubt about what outcome is appropriate for any given **case in scope**.

Single-Sourcing. Suppose the appropriate outcome for a *rainy*-but-*not-cold* day (*unlined raincoat*) were to change (say, to *umbrella*). In the intersection-style decision table above, there is one and only one place, one quite obvious, to make that change.

Important:

> *Single-sourcing is always a best practice in* **business rules**.

The reason is simple – things change constantly. So modifying any part of decision logic should always be intentional (and traceable), not accidental or haphazard.

1.4.1 Row-or-Column-Style Decision Tables

Achieving these important goals in the **row-or-column-style** is not quite so straightforward – *at least without specialized software support*.

Decision Rules. To illustrate potential problems, deliberate mistakes have been made in Figure P1-13. Keep in mind these mistakes are relatively easy to spot because the decision table is so small.

Figure P1-13. Mistakes of Redundancy and Omission in the One-Rule-Per-Row Style Decision Table

Is it cold?	Is it rainy?	*What coat should be worn?*
yes	yes	*lined raincoat*
	no	*wool overcoat*
	yes	*lined raincoat*
no	yes	*unlined raincoat*

Completeness. The first deliberate mistake in Figure P1-13 is the omission of the last row in Figure P1-4 (indicating no coat should be worn if it is not cold and not rainy). Unlike the intersection style, there is no blank cell to warn you; *the row just isn't there*. So it's not so easy to see that the decision logic is incomplete.

Single-Sourcing. The second deliberate mistake in Figure P1-13 is the repetition of the decision rule for a cold, rainy day. The decision rule is not single-sourced, which leaves the door open to integrity (correctness) problems. Redundancy like this is not so easy to spot either.

1.4.2 Challenges for the Row-or-Column Style

Intersection Cases. One challenge for the row-or-column-style is that *intersection cases*, rather than elemental cases, need to be exhaustive and non-redundant. (In technical jargon ensuring strict, logical uniqueness of intersection cases is called **single-hit**.)

Since the style does not exploit the physical matrix in that regard, specialized software is inevitably required to support it. Surprisingly, some software tools don't provide it.

Elemental Cases. In the row-or-column-style, elemental cases for all considerations *cannot be listed non-redundantly*. In fact, the elemental cases for every consideration except one are listed redundantly in order to set up intersection cases. (Use of the special symbol *does not matter* (dash) often reduces the total number of repetitions. Refer to Appendix 7.)

In general, a large number of elemental cases will not be single-sourced. Refer to Figures P1-6 and P1-9 for illustrations. Managing such redundancy can prove non-trivial. Refer to Part 4.1.3 for discussion.

1.4.3 Clarification Regarding Outcomes

Two points concerning outcomes on which all the styles discussed above do *not* differ:

1. One or more **potential outcomes** might be absent from a decision table. A decision table is *not* required to use the complete set of potential outcomes.
2. Outcomes need not be unique within a decision table. Any potential outcome can appear multiple times – that is, as the outcome for *more* than one decision rule.

In other words, no decision table laid out in any of the styles above ever needs to be exhaustive or non-redundant with respect to outcomes. The styles *cannot* be differentiated on that basis.

1.5 THE MEANING (SEMANTICS) OF DECISION TABLES

No **decision table** is complete until its meaning (semantics) has been fully exposed. Otherwise, opportunities for misinterpretation abound.

The meaning of a decision table involves two core aspects:

Aspect of Meaning 1. The *individual* meanings of terms and wordings – i.e., the business vocabulary – used for the decision table.

> A decision table should always be based on a **structured business vocabulary (concept model**). Refer to Part 4.1 for discussion and an example.
>
> Part 4.1 also highlights the critical importance of continuing alignment between question and outcomes, a central TableSpeak guideline.

Aspect of Meaning 2. The *collective* purpose and relevance (semantics) of the decision table.

> Three **TableSpeak** approaches for explicitly communicating the collective semantics of a decision table are discussed below in this Part, along with advantages and possible disadvantages of each. The discussion focuses on two items of special importance, *name* and *scope*.
>
> Other important items (discussed in Part 3) include **defaults**, **exceptions**, and **restrictions**.

Without adequate clarification of both aspects of meaning, a decision table cannot serve its purpose very well, whether for actually running the business or as a requirement for IT development.

Remember, as is true for all **business rules**, you can never be sure who will eventually view a decision table, or for what purpose they might try to use it.

1.5.1 The Name of a Decision Table

Common industry practice is to name a decision table roughly after what it concerns. The decision table in Figure P1-1, for example, concerns the choice of coats, so it might be called *Coat Table*.

TableSpeak, however, recognizes the central importance to the business of the question that a decision table answers. That question is fundamental to what the decision table represents (i.e., its meaning).

> TableSpeak *strongly recommends using the question as the table's name.*

Instead of *Coat Table*, for example, **TableSpeak** would recommend *What coat should be worn?*.

1.5.2 The Scope of a Decision Table

Addressing scope is essential for analyzing **operational business decisions** and representing the **decision logic** that supports them.[5]

Delineating scope for operational business decisions is often not a trivial matter. With respect to decision tables, misuse is almost guaranteed without adequate specification and communication of scope.

For example, suppose the decision table in Figure P1-1 was created to represent decision logic applicable to just *females*, and just to *San Francisco*. Use of the decision logic for males or any other city would therefore be unwarranted and very likely incorrect.

[5] The practitioner is urged to carefully review the important discussion about scope in Appendix 5 of the companion Primer on **DecisionSpeak**. http://www.brsolutions.com/IPSpeakPrimers

Such use would represent **inductive reasoning**, something generally inappropriate for business rules.

Unless the scope of a decision table is universal within a business (not impossible but seldom the case) some explicit **scope item(s)** should be expressed for the decision table.

The scope items for the decision table in Figure P1-1, for example, could be expressed as follows. (An implicit 'and' is always assumed between scope items.)

- gender: *female*
- city: *San Francisco*

Scope items also serve as a TableSpeak technique to achieve the crucial objective of keeping decision tables as simple as possible.

For example, treating *gender* and *city* as considerations for the decision table in Figure P1-1 is pointless (at least at the present time). Doing so would simply add complexity – and that complexity would compound as the size of the decision table grows.

A final observation: Scope items do not preclude modifying a decision table to broaden its coverage. When the scope is broadened the relevant scope items should simply be dropped or revised as appropriate.

Now let's examine the three **TableSpeak** approaches for explicitly communicating the collective semantics of a decision table

1.5.3 Approach to Semantics 1: Wrapper Rule Statement

A concern in **rule** management is often that decision tables participate fully in a larger body of guidance (business rules) that also includes a significant number of textual business rule statements.

In this circumstance writing a **wrapper rule statement** for the decision table often proves the best approach. That way each decision table has a specific textual 'handle' in and among all the other rule statements.

The wrapper rule statement for a decision table indicates what the decision table represents and how it should be interpreted – i.e., how to 'read' the decision table.

For example, an appropriate wrapper rule statement for the decision table in Figure P1-1 might be expressed as follows. Note that the two scope items have been carefully included as qualifications.

> Wrapper Rule Statement: *The coat for a female person to wear in San Francisco must be as in the decision table 'What coat should be worn?'.*

Suppose these scope items can be suitably documented elsewhere and still remain highly visible whenever the decision table is used (a bit doubtful). A shorter 'stub' version might be:

> Revised Version: *The coat to wear must be as in the decision table 'What coat should be worn?'.*

1.5.4 Approach to Semantics 2: Embedded Semantics

A second approach aims at making the decision table itself as self-explanatory and as self-contained as possible. This approach involves embedding its collective semantics directly into it. No wrapper rule statement is necessary. Figure P1-14 illustrates.

Figure P1-14. Decision Table with Embedded Semantics

		Is it rainy?	
	What coat should be worn? Scope Items: *1. gender: female* *2. city: San Francisco*	***yes***	***no***
Is it cold?	***yes***	*lined raincoat*	*wool overcoat*
	no	*unlined raincoat*	*none*

In this approach both of the following have been directly embedded within the decision table:

- The question the decision table addresses (which also serves as the name of the decision table).
- The two scope items.

This approach serves quite well for this simple example. As additional semantics are added, however, such representation can become increasingly cluttered.

1.5.5 Approach to Semantics 3: Decision Box

A third approach aims toward providing a focal point for managing all the interrelated semantics of a more complex decision table.

This approach involves creating a paired **decision box** for each decision table, which always appears whenever the decision table itself appears. Again, no wrapper rule statement is necessary. Figure P1-15 illustrates.

This decision table features a paired decision box providing the table's collective semantics:

- The question the decision table addresses (which also serves as the name of the decision table).
- The two scope items.

The usefulness of a decision box becomes more and more apparent as the number and complexity of specifications related to the

Figure P1-15. Decision Table with Paired Decision Box

What coat should be worn?
Scope Items:
1. gender: female
2. city: San Francisco

	Is it rainy?	
	yes	***no***
Is it cold? ***yes***	*lined raincoat*	*wool overcoat*
no	*unlined raincoat*	*none*

decision table increase. As mentioned earlier, such specifications can include **defaults**, **exceptions** and **restrictions**.

Besides high visibility for these aggregate semantics, the decision box also provides a focal point for managing all such specifications *as a unit* (i.e., **single-sourced**). These specifications can prove quite difficult to coordinate on an individual basis apart from the decision table.

To highlight the importance of the aggregate semantics, sometimes the best approach is to put the decision table *inside* its decision box. Figure P1-16 illustrates.

Figure P1-16. Decision Box with Embedded Decision Table

Decision: What coat should be worn?

Scope Items:
1. gender: female
2. city: San Francisco

Is it cold? \ *Is it rainy?*	***yes***	***no***
yes	*lined raincoat*	*wool overcoat*
no	*unlined raincoat*	*none*

PART 2: STYLES OF DECISION TABLES

2.1 DECISION TABLES WITH ONE CONSIDERATION

Most **decision tables** are not as simple or so small as those examined so far. As subsequent sections discuss, for example, many decision tables have more than two **considerations**.

Even a decision table with only one consideration might include complications. For example it might:

- Address *many* **elemental cases** for a given **consideration**.
- Specify some elemental cases as *brackets* or *ranges*.

Figure P2-1 illustrates using a new **operational business decision**, *What should be charged for shipping an order?*.

Figure P2-1. Decision Table with One Consideration

What should be charged for shipping an order?	
zip code	*shipping cost*
00401	$29
04401	$26
04402	$21
04730 to 04739	$35
04740	$19
~~	~~
~~	~~
99928 to 99929	$48
99950	$36

Observations

- The decision table in Figure P2-1 has just one consideration, *zip code*. The appropriate **outcome**, *shipping cost*, is given for each elemental case of *zip code*. Since there is only one consideration, the decision table does not address any **intersection cases**.
- Many elemental cases are given (for *zip code*) in the decision table.

  A consideration with more than two or three elemental cases is called a **many-case consideration**.
- Some elemental cases in the decision table are specified as brackets or ranges (e.g., *04730 to 04739*).

A consideration with many elemental cases often produces a sizable table. For example, there are approximately 43,000 *zip codes* in the United States.[7]

To facilitate use, breaking the set of cases into sections, or even multiple tables, is often desirable.

Is the decision table in Figure P2-1 complete? Obviously many *zip codes* seem to be missing.

> *Aside:* For the purpose of this Primer, squiggly lines in any cell of a decision table simply indicate there are too many cases to show. A squiggly line has no inherent

[7] http://www.carrierroutes.com/ZIPCodes.html, available March, 2013

meaning (semantics) and would not be used in actual practice.

2.2 INTERSECTION-STYLE DECISION TABLES WITH TWO CONSIDERATIONS

Most **decision tables** involve more than one **consideration**. For example, an additional consideration for the **operational business decision** *What should be charged for shipping an order?* illustrated in Figure P2-1 might be *weight*.

In the **intersection style**, **elemental cases** of this second consideration can be displayed along the other axis. Figure P2-2 illustrates.

Observations

- The decision table in Figure P2-2 now includes a second consideration, *weight.* Five of its elemental cases (all ranges) are shown. Many more elemental cases for the consideration might be included in the decision table.
- For a given *zip code* (or range of *zip codes*) the *shipping cost* is now differential with respect to *weight*. An appropriate **outcome** (*shipping cost*) is shown for each **intersection case**.
- The number of **decision cells** has now increased by at least a factor of five (and probably much more).

2.3 MULTI-TABLE REPRESENTATION OF DECISION LOGIC

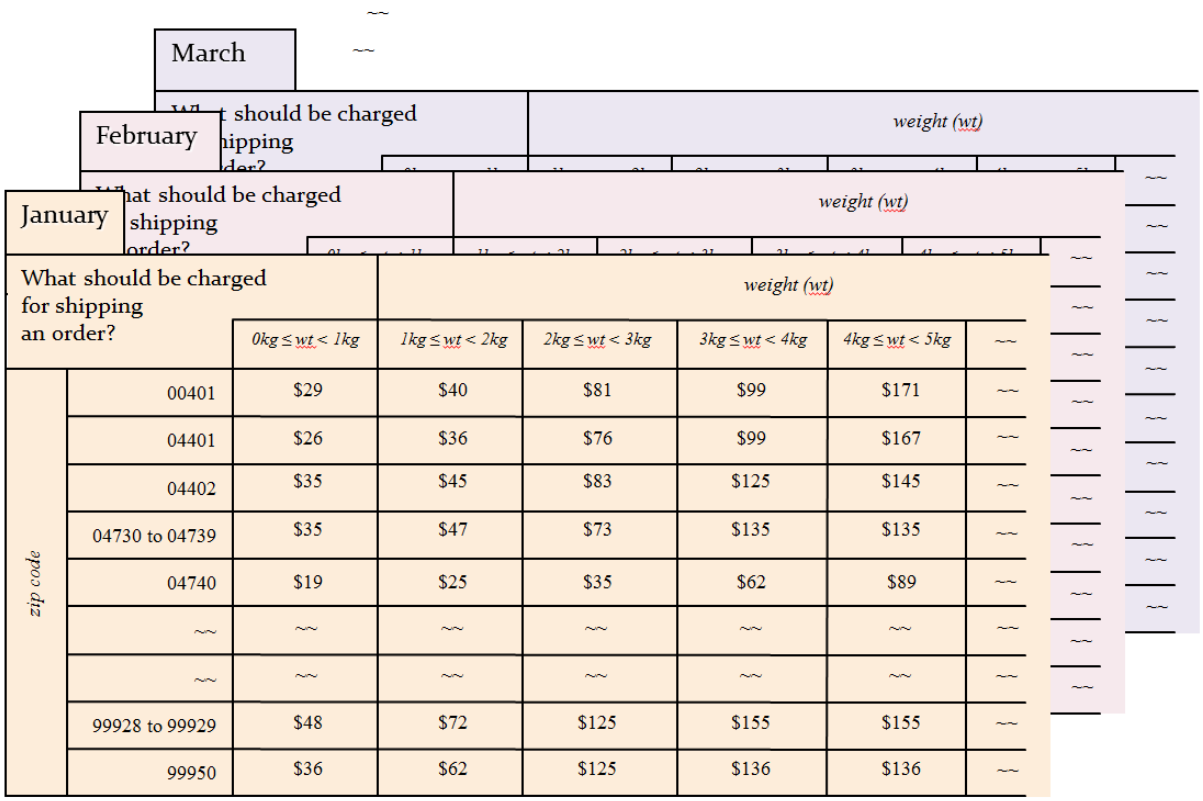
Some **decision tables** involve more than two **many-case considerations**. For example, an additional consideration for the **operational business decision** *What should be charged for shipping an order?* illustrated in Figure P2-2 might be *month*.

In the **intersection style**, **elemental cases** of this third consideration cannot be displayed in a two-dimensional matrix. A third dimension is required.

Figure P2-2. Intersection-Style Decision Table with Two Considerations

What should be charged for shipping an order?		*weight (wt)*					
		0kg ≤ wt < 1kg	*1kg ≤ wt < 2kg*	*2kg ≤ wt < 3kg*	*3kg ≤ wt < 4kg*	*4kg ≤ wt < 5kg*	~~
zip code	00401	$29	$40	$81	$99	$171	~~
	04401	$26	$36	$76	$99	$167	~~
	04402	$35	$45	$83	$125	$145	~~
	04730 to 04739	$35	$47	$73	$135	$135	~~
	04740	$19	$25	$35	$62	$89	~~
	~~	~~	~~	~~	~~	~~	~~
	~~	~~	~~	~~	~~	~~	~~
	99928 to 99929	$48	$72	$125	$155	$155	~~
	99950	$36	$62	$125	$136	$136	~~

Figure P2-3. Multi-Table Representation of Three Considerations



The easiest way to think of a three dimensional array is simply as multiple tables. Figure P2-3 illustrates.

Observations

- The **multi-table** representation of the decision logic in Figure P2-3 is based on the third consideration, *month.*
- Three elemental cases of *month* are shown (*January, February,* and *March*), each associated with a separate table. Tables for the nine remaining *months* have not been shown.
- For a given *zip code* (or range of *zip codes*) and a given *weight* bracket, the *shipping cost* is now differential with respect to *month*. The number of **decision cells** has increased by a factor of twelve.

2.3.1 The Multi-Table Threshold for Decision Logic

The example illustrated in Figure P2-3 has crossed an important threshold for the **intersection style**, which in **TableSpeak** is called the **multi-table threshold**.

At *three* many-case considerations, multiple tables are apparently required to represent decision logic. ("Apparently" is simply a reminder that the easiest way to visualize a three-dimensional array is as multiple tables.)

What is the simplest way to represent **multi-table** *decision logic?*

The best consideration to use as the basis for **multi-table** representation is generally the consideration with the fewest number of elemental cases.

That's one reason *month* was selected as the basis in Figure P2-3 – it has only 12 elemental cases, compared to 10,000s for *zip code* and possibly 100s for *weight*.

The elemental cases of the selected consideration should also:

- Be familiar to business people.
- Provide a good starting point to obtain an outcome for any given intersection case.

The consideration *month* rates highly on these criteria too.

How many tables are required in a **multi-table** *representation?*

The number of two-dimensional tables required to represent three considerations is equal to the number of elemental cases for the third consideration. *Month* for example has 12.

Can the decision logic for an operational business decision involve more than three many-case considerations?

Yes, but consider *very* carefully the number of decision cells such representation involves.

Suppose there are four many-case considerations that have *m, n, p,* and *q* number of elemental cases respectively. The number of decision cells is multiplicative – specifically, $m \times n \times p \times q$. That's probably *a very large number!*

Beyond three many-case considerations, the practitioner is well-advised to reconsider the operational business decision by analyzing the following questions:

- Can some portion(s) of the outcomes be computed by some formula?
- Can the operational business decision be decomposed into subdecisions? Refer to the *DecisionSpeak Primer* for discussion.[8]
- For the given decision, is it really all that important to manage the decision logic *in words*?[9]
- Considering the people-cost of managing a very large decision table, is the selectivity afforded by $m \times n \times q \times p$ outcomes really worthwhile to the business?

The last point is especially important.

Sometimes it is easy to forget that decision logic not only must be captured and represented, but also *managed* over time (kept up-to-date and **anomaly**-free). There are no free lunches(!).

Ask: Is the complexity *really* cost-effective for the business? Is managing the selectivity *really* worth the overhead it entails?

The **row-or-column-style**, incidentally, offers *no relief whatsoever* on this need-to-manage issue. Venture beyond three many-case considerations *only very carefully*.

[8] http://www.brsolutions.com/IPSpeakPrimers

[9] For example as opposed to mathematical models. For explanation refer to Appendix 2 of the *DecisionSpeak Primer:* http://www.brsolutions.com/IPSpeakPrimers

2.4 FEW-CASE AND EMBEDDED CONSIDERATIONS IN INTERSECTION-STYLE DECISION TABLES

A **few-case consideration** is a consideration with only two or three **elemental cases**. For that reason a few-case consideration is sometimes called a **simple consideration**. Examples:

1. The few-case consideration *Is it cold?* in Figure P1-1 produces the two elemental cases:
   - *Yes, it's cold.*
   - *No, it's not cold.*
2. The few-case consideration *category of customer* might produce three elemental cases:
   - *platinum*
   - *gold*
   - *silver*

How are few-case considerations handled in creating **decision tables**?

Decision logic with a reasonable number of few-case considerations can be handled in **intersection style** decision tables. The intersection-style decision table in Figure P1-1 for example included two few-case considerations: *Is it cold?* and *Is it rainy?*.

What is the highest number of few-case considerations considered *reasonable*? The answer depends on several factors:

- Does the **decision logic** include more than *five* few-case considerations (the **row-or-column threshold**)?

  If so, the **row-or-column-style** is generally more suitable. Refer to Part 2.5 for discussion.

- How many **many-case considerations** are present?

  The more many-case considerations are present (if any), the lower the number of few-case considerations that can be easily handled. The following discussion explains.

Let's return to the **operational business decision** *What should be charged for shipping an order?* as first given in Figure P2-1. At that point, the decision logic had only a single many-case consideration, *zip code*.

Now let's add the few-case consideration *kind of packaging*, which is known to have only two **elemental cases** – *box* and *tube*. For a given *zip code,* selective outcomes for this new consideration are shown in Figure P2-4.

Figure P2-4. Selective Outcomes for a Decision Table with One Many-Case Consideration and One Few-Case Consideration

What should be charged for shipping an order?		
	kind of packaging	
zip code	box	tube
00401	$29	$35
04401	$26	$31
04402	$21	$24
04730 to 04739	$35	$45
04740	$19	$19
~~	~~	~~
~~	~~	~~
99928 to 99929	$48	$37
99950	$36	$37

Observations

- The decision table in Figure P2-4 includes a second consideration, *kind of packaging*.
- The two elemental cases of this few-case consideration, *box* and *tube*, appear at the top of the two rightmost columns.
- Because the decision table uses both dimensions of the matrix (the vertical and the horizontal), the decision table is intersection-style.
- The columns for the two elemental cases produce differential **outcomes** for *shipping cost* based on *zip code*.

Caution

Always be alert to the **completeness** *of elemental cases – i.e., whether the set of elemental cases is* **exhaustive**.

Unlike *Is it rainy?,* which can produce only *yes* and *no*, the new consideration in Figure P2-4, *kind of packaging*, is not intrinsically binary.

Suppose the **structured business vocabulary** (**concept model**) indicates *reinforced envelop* to be an additional *kind of packaging*. If so the decision table in Figure P2-4 is incomplete. Refer to Part 5.2 for additional discussion of quality concerns for elemental cases.

To continue developing the example let's reintroduce *weight* as first given in Figure P2-2. How would the few-case consideration *kind of packaging* be handled alongside the two many-case considerations *zip code* and *weight* in the intersection style?

One way to handle the three considerations is **multi-table**, as illustrated in Figure P2-5.

Figure P2-5. Decision Table with Two Many-Case Considerations and One Few-Case Consideration Using Multi-Table Representation

Figure P2-6. Decision Table with Two Many-Case Considerations and One Few-Case Consideration Using Embedding

What should be charged for shipping an order?	*weight (wt)* and *kind of packaging*										
	0kg ≤ wt < 1kg		*0kg ≤ wt < 1kg*		*0kg ≤ wt < 1kg*		*0kg ≤ wt < 1kg*		*0kg ≤ wt < 1kg*		~~
zip code	*box*	*tube*	*box*	*tube*	*box*	*tube*	*box*	*tube*	*box*	*tube*	~~
00401	$29	$35	$40	$45	$81	$99	$99	$141	$171	$201	~~
04401	$26	$31	$36	$41	$76	$93	$99	$137	$167	$157	~~
04402	$35	$43	$45	$55	$83	$95	$125	$145	$145	$145	~~
04730 to 04739	$35	$45	$47	$53	$73	$87	$135	$138	$135	$131	~~
04740	$19	$19	$25	$27	$35	$38	$62	$59	$89	$72	~~
~~	~~	~~	~~	~~	~~	~~	~~	~~	~~	~~	~~
~~	~~	~~	~~	~~	~~	~~	~~	~~	~~	~~	~~
99928 to 99929	$48	$37	$72	$49	$125	$101	$155	$101	$155	$109	~~
99950	$36	$37	$62	$48	$125	$98	$136	$99	$136	$110	~~

2.4.1 Embedded Cases

Another way to handle the three considerations in the example above is by *embedding* the few-case consideration.

In **embedding**, elemental cases for one consideration are repeated within every elemental case of another consideration. Figure P2-6 revises the decision table in Figure P2-5 using embedding.

Observations

- The decision table in Figure P2-6 embeds the few-case consideration, *kind of packaging*, within the many-case consideration *weight*.
- The two elemental cases of this few-case consideration, *box* and *tube*, are *repeated* (embedded) within each elemental case of *weight*.
- In contrast to Figure P2-5, representation of the decision table is no longer **multi-table**.
- The decision table gives differential outcomes for *shipping cost* based on *zip code*, *weight,* and *kind of packaging*.

The consideration *weight* was chosen for the embedding of *kind of packaging* because it presumably has fewer elemental cases than *zip code*. That choice produces the least amount of overall replication, an important goal.

Cautions

1. *In the intersection style, TableSpeak advises embedding only few-case considerations.*

   A consideration with four or more elemental cases (i.e., a many-case consideration) involves so much repetition that the resulting decision table is unlikely to be business-friendly.

2. *Redundancy of elemental cases always produces potential integrity (correctness) concerns.*

   For example, suppose the elemental case *box* needs to be revised to *carton.* Since this

elemental case is now not **single-sourced**, the change must be made in multiple places. If not changed consistently and comprehensively, the integrity of the decision table is damaged.

3. *Use of embedding in the intersection style degrades the natural (i.e., structural) resistance of the style to this kind of* **anomaly**.

2.4.2 Selection of the Best Representation for Decision Logic

As discussed above, your choice of format using the intersection style can be extended by multi-table representation and/or by embedding some few-case consideration(s).

Refer to Part 5.1 for a comprehensive set of TableSpeak recommendations in this regard. Part 5.1 also:

- Reviews the three major style-related thresholds in TableSpeak: **multi-table threshold**, **row-or-column threshold**, and **complexity threshold**.
- Indicates when the row-or-column-style might work best. Part 2.5 examines that style more closely.

2.5 ROW-OR-COLUMN-STYLE REPRESENTATION OF DECISION TABLES

An important threshold in representing **decision logic** is reached when the number of **considerations** *exceeds five.*

At this threshold the **intersection style** can no longer be used effectively (at least without appropriate automated support). The **row-or-column-style** needs to be used instead.

This important threshold in **TableSpeak** is called the **row-or-column threshold**.

As Part 5.1 indicates, this threshold applies for *any* combination of **few-case** and **many-case** considerations.

Refer to Part 1 for a refresher on the row-or-column-style and how it differs from the intersection style. Keep in mind that the different styles are about *visualizing* **decision rules**; they are not about data management or database schemes.

The row-or-column-style always unites (concatenates) **elemental cases** in a series, which are represented in rows or columns.

- If in rows, the format is **one-rule-per-row**.
- If in columns, the format is **one-rule-per-column**.

As explained in Part 1, these formats are only superficially different.

The row-or-column-style is less friendly to business people and is more prone to **anomalies**. TableSpeak recommends its use only when absolutely necessary.

> *The origin of the row-or-column-style was primarily in software development and programming where binary considerations (branch points in flowcharts) abound.*

Refer to Appendix 8 for additional background about the style.

Figure P2-7. A One-Rule-Per-Row Decision Table for the *What is the right delivery method for an order?* Decision

What is the right delivery method for an order?							
rush order?	*any fragile item?*	*any specialty item?*	*any high-priced item?*	*any item involving hazardous materials?*	*category of customer?*	*destination of order?*	***delivery method for order***
no	*no*	*no*	*no*	*no*	*silver*	—	**picked up by customer**
yes	*yes*	*no*	*no*	*yes*	*gold*	*local*	**shipped by normal service**
yes	—	—	—	*yes*	*platinum*	*remote*	**shipped by premium service**

Figure P2-7 illustrates the one-rule-per-row style for a new **operational business decision**, *What is the right delivery method for an order?* (Refer to Appendix 3 for the same decision logic formatted in the one-rule-per-column style.)

This example has purposely been kept very simple to facilitate discussion and comparison. Assume many more cases (rows) are to be added later.

The decision table in Figure P2-7 is organized as follows:

- Seven few-case considerations are listed along the top row.
- Three **intersection cases** are shown in each of the three lower rows.
- The appropriate **outcome** for each of the three intersection cases, the particular *delivery method for order*, is shown at the right of each row, in the rightmost column.
- Each complete row holds one **decision rule**. These rules could be verbalized individually as in Appendix 4.

2.5.1 *Does Not Matter* and Subsumed Elemental Cases

Rather than showing a particular elemental case, some cells in Figure P2-7 show a dash. A dash in this context means *does not matter*.

In other words, the outcome for that row is the same no matter what elemental case is indicated for the given consideration. Refer to Appendix 7 for additional discussion of the special symbol *does not matter* (dash).

A decision rule that includes a *does not matter* (dash) for one or more of its considerations is literally a **general rule** since it covers more than one intersection case. A general rule is simply any decision rule that covers multiple (often many) cases.

A *does not matter* (dash) is a means by which the number of intersection cases in a decision table can be kept as low as possible. In the one-rule-per-row style that means the fewest number of rows.

A *does not matter* (dash) can arise in a decision table in either of two ways:

- Direct specification by the people creating or maintaining the decision table.
- A **subsumption** is discovered and the decision table is *contracted* – i.e., some row(s) are eliminated (in the one-rule-per-row style).

Refer to Part 5.9 for discussion of subsumption. Incidentally, automated tools can provide useful support in this area.

Cautions

1. *The intended meaning of 'does not matter' can be unclear.*

   For example, which of the following does *does not matter* (dash) mean?

   (a.) 'does not matter' *and may be unknown*, or

   (b.) 'does not matter' *but must be known.*

   In TableSpeak a *does not matter* (dash) always means the former. If something *must* be known, a **consideration restriction** should be specified.

   Refer to Part 5.8 for discussion.

2. *Manipulation of a decision table based on subsumption is not without certain dangers – original business intent can be lost.*

   Refer to Part 5.11 for discussion.

2.5.2 Using Row-Or-Column-Style Decision Tables Effectively

As previously discussed, the row-or-column-style always unites (concatenates) **elemental cases** in a series, which are represented in rows or columns.

The elemental cases for at most only one consideration can be non-redundant.

The entire set of elemental cases for the second consideration must potentially be repeated within *each* elemental case of the first consideration; the entire set of elemental cases for the third consideration within *each* elemental case of the second consideration; and so on.

At the extreme every possible combination of elemental cases for all considerations is represented in rows or columns. (Not all such **intersection cases** always appear, however, because of subsumption.)

The row-or-column-style represents radical **embedding**.

The following important concerns arise from this fundamental characteristic of the row-or-column-style.

Uniqueness of Intersection Cases. *It is absolutely essential that all rows or columns in the row-or-column-style (i.e., all intersection cases) be strictly unique (non-redundant).*

If not *unique*, important **anomalies** (**conflicts**, **multiple outcomes**, and **redundancies**) can occur. Refer to Part 5.5 for discussion. For row-or-column-style decision tables beyond very modest size, ensuring uniqueness *requires software support*.

In technical jargon ensuring strict, logical uniqueness of intersection cases is called **single-hit**. The potential for anomalies if

'single-hit' is not enforced has been recognized for decades.[10]

Exhaustive Intersection Cases *It is also absolutely essential that the set of intersection cases in a row-or-column-style decision table be represented exhaustively.*

If not *exhaustive*, **omissions** – missing decision rules – can occur. Refer to Part 5.6 for discussion of this kind of **anomaly**.

At issue is the *completeness* of decision logic. Appendix 5 assesses the completeness of the decision table in Figure P2-7. (It's not complete at all!)

Single-Sourcing of Elemental Cases. *Redundancy of elemental cases always produces potential integrity (correctness) concerns.*

Whenever an elemental case is repeated (not **single-sourced**), any modification must be made in multiple places. If not changed consistently and comprehensively, the integrity of the decision table is diminished.

In the row-or-column-style the redundancy of elemental cases can be *massive*.

Size. *In general, where more than one many-case consideration is present, the row-or-column-style is practical only if every many-case consideration after the first one has a relatively small number of elemental cases.*

Not following this guideline can result in huge numbers of rows or columns. Refer to Appendix 6 for an illustration.

Software that supports contraction and expansion of decision tables based on **subsumption** is generally essentially for such decision logic. Refer to Part 5.9 for a simple illustration of contraction/ expansion.

For both the row-or-column-style and the intersection style, *overall size* is dictated by the number of intersection cases.

The number of **decision cells** *must necessarily be the same for the same decision logic rendered in either style – it's the same decision logic just visualized in different formats.*

Business Friendliness. *The row-or-column-style for decision tables tends to be less intuitive and harder to use than the intersection style.*

IT developers often like the row-or-column-style, but business people and business analysts often struggle with it.

In larger decision tables, any particular intersection case of interest can be difficult to find, especially if the rows or columns are not kept well-ordered.

Software Support. *The row-or-column-style almost unavoidably requires automated support for decision tables of any size.*

Intersection-style decision tables are a bit better in that respect, but also require automated support as the size and complexity of decision tables increase.

At some point, the need for automated support has to be taken simply as a fact of life.

[10] from at least the time of the Codasyl report, *A Modern Appraisal of Decision Tables, Report of the Decision Table Task Group*, ACM, New York, 1982.

2.6 AVOIDING SEQUENCE IN DECISION TABLES

Decision logic and process models should be viewed as serving fundamentally different purposes.

- Decision logic provides the best or optimal answers to business questions. Decision logic does not flow. It should not be viewed as actually *doing* (transforming) anything.
- Process models indicate how tasks should be performed in some meaningful sequence. Process models always flow. They always actually *do* (transform) something.

In short, decision logic should be **declarative**; process models (by definition) should be **procedural**.

Related **TableSpeak** guidelines for designing **decision tables** are therefore:

1. *Never use a decision table simply to indicate what to do next.*

   Decision tables should never include any explicit actions or refer to any processes or procedures. All **considerations** should represent business factors; all **outcomes** should be answers to some business question.

   Process models, logic trees, and flowcharts work perfectly well to represent the procedural aspects of business capabilities.

2. *Never design a decision table such that it matters in what order:*

   - *considerations are evaluated.*
   - *rows are evaluated.*
   - *columns are evaluated.*

   Sequence should simply never matter in reading or interpreting a decision table.

Decision tables built following these TableSpeak guidelines are best for addressing the inherent complexity of **operational business decisions**. Such decision tables also inevitably prove the most:

- business-friendly
- re-usable
- platform-independent
- easy to change (agile)
- manageable

2.6.1 Best Practices

The simple example presented in Part 4.2 illustrates how a decision table can be used inappropriately to represent procedural logic. It also explains what to do about it.

The **row-or-column-style** of decision tables is particularly susceptible to misuse in this regard.

- The left-to-right layout of the typical **one-rule-per-row** decision table is quite easily misused to represent flow or sequence, often unknowingly. (The same is true for the typical top-to-bottom layout of **one-rule-per-column** decision tables.)
- The row-or-column style presents no natural disincentives against *empty* cells for **elemental cases**, especially after the first consideration.

  As a result it is quite easy for a subsequent consideration to depend fully on just one elemental case of the previous consideration. *Such representation is basically just flowcharting in tabular form.*

Part 4.3 discusses how to turn a sequential dependency within a decision table into a logical one. It's actually quite easy. Although the conversion is illustrated using decision tables, the underlying principle is one that applies to *all* **business rules**.

PART 3: THE INTEGRITY OF DECISION TABLES

3.1 OVERVIEW OF INTEGRITY CONCERNS

Three basic reasons for diminished integrity (correctness) in **decision logic** are:

- *Invalid* **decision rules**.

  This business problem can be resolved only by knowledge workers and business analysts working under an appropriate methodology.

- *Structural deficiencies and lack of adequate support that permit* **anomalie**s *to arise in* **decision tables**.

  As discussed in Part 2 and related Appendices, potential anomalies can be controlled by careful design and if available, appropriate software.

- *Lack of adequate controls over management and revision of decision tables such that their meaning (semantics) is compromised.*

  Part 3 of the Primer focuses on this third problem, which centers on *decision table management*.

Pragmatic protection of meaning (semantics) is a distinguishing characteristic of **TableSpeak**. Achieving consistency of **operational business decisions** over time, and the lowest possible cost-of-management, requires focused techniques for this area.

Unfortunately, coordination of integrity has been widely neglected in the field. Software vendors are notorious in that regard.

Remember, *there are no silver bullets!* You simply can't forget everything you know (or should know) about **business rules** when you do decision tables.

Central to the area is thoughtful, deliberate treatment of **exceptions**, **defaults** and **restrictions**. Each is examined closely in subsequent sections in this part of the Primer.

Exceptions, defaults and restrictions are part of the semantics (meaning) of a decision table. They should be managed within or alongside it.

> *For this reason TableSpeak generally recommends* **decision boxes** *to accompany decision tables, especially for decision tables of greater complexity.*
>
> Refer to Part 1 for an introduction to decision boxes.

All TableSpeak guidelines for integrity are based on the assumption that decision logic *will* change, often quite rapidly.

The best design is therefore one that caters to such change, always doing so in the manner friendliest to business people and business analysts.

3.1.1 TableSpeak Guidelines for Elemental Cases and Outcomes

The most basic area of integrity for decision tables pertains to composing **elemental cases** and **outcomes**. Refer to Parts P5.2 and P5.3 and to P5.4, respectively, for important guidelines.

Table P3-1. Kinds of Restriction in TableSpeak

kind of TableSpeak restriction	*effect on decision tables*
relevance restriction	Always either: • precludes providing any outcome(s) for some case(s) – *preemption*. • warns that if any outcome is provided for some case(s), the outcome cannot be considered necessarily valid – *caveat*.
consideration restriction	Always precludes certain ways in which considerations can be combined.
outcome restriction	Always limits certain cases to a particular ('fixed') outcome or to some subset of all potential outcomes.

One critically important TableSpeak convention:

> *Every cell for an intersection case in a decision table must have something in it.*

TableSpeak says it this way:

> *Empty is not an option. Empty hides meaning – semantics.*

3.1.2 Restrictions for Decision Tables

A **restriction** is *a business rule that directly governs the integrity (correctness) of a decision table*.

Restrictions apply to the content of almost any decision table. Identifying, specifying and managing restrictions are a critical part of decision table management. Failure in this regard can exact a high price.

Table P3-1 lists the three fundamental kinds of restriction that can be placed on decision tables. Each is discussed and illustrated in later sections of this part of the Primer.

Observations

- These three kinds of restriction correspond *exactly* to the three kinds of decision dependencies in **DecisionSpeak**: **relevance dependency**, **consideration dependency**, and **outcome dependency**, respectively.[11] DecisionSpeak and TableSpeak fit hand-in-glove with each other.
- Between them, these three kinds of restriction cover everything in a decision table:
  1. The question (via relevance restrictions).
  2. The consideration(s) – and through them, elemental cases and intersection cases.
  3. Outcomes.

  Literally, there is nothing else in decision tables to restrict.
- A restriction often affects multiple cells in a decision table. Defining a restriction not only allows **single-sourcing** of the related business intent, but also supports its faithful retention and consistent application as the decision table undergoes modifications over time.

[11] *Decision Analysis: A Primer – How to Use DecisionSpeak™ and Question Charts (Q-Charts™):* http://www.brsolutions.com/IPSpeakPrimers

3.2 EXCEPTIONS

The **outcome** for any given **case** is usually determined by a standard or typical set of **considerations**. If not, then the case is an *exceptional case*.

An **exceptional case** is *a case in scope that does not use any of the considerations of a standard (normal) case.* Each exceptional case presumably uses some other consideration(s).

Suppose *Halloween* is an exception case for the *What coat should be worn?* **decision table** originally presented in Figure P1-1. *Halloween* is a *special event* or *holiday*. Consequently the 'normal' considerations *Is it cold?* and *Is it rainy?* cannot be used to determine an appropriate outcome for that case.

Since the exceptional case *is* within scope an outcome still must be provided for it. Suppose the appropriate outcome is *costume*. Figure P3-1 illustrates specification for the exceptional case as embedded semantics.

Note that an exception such as the one involving *Halloween* is actually being handled by a **decision rule** – just not one in the decision table proper. As always in **business rules**, an exception to a rule is simply another rule.

Figure P3-2 illustrates the same specification using a **decision box**.

As these decision tables illustrate, **TableSpeak** strongly recommends handling exceptional

Figure P3-1. Specifying an Exception as Embedded Semantics

What coat should be worn? **Scope Items:** *1. gender: female* *2. city: San Francisco* **Exception:** *1. Halloween: costume*	*Is it rainy?*	
	yes	***no***
Is it cold?		
yes	*lined raincoat*	*wool overcoat*
no	*unlined raincoat*	*none*

Figure P3-2. Specifying an Exception Using a Decision Box

What coat should be worn?

Scope Items:
1. gender: female
2. city: San Francisco

Exception:
1. Halloween: costume

	Is it rainy?	
	yes	***no***
Is it cold?		
yes	*lined raincoat*	*wool overcoat*
no	*unlined raincoat*	*none*

cases *apart* from the **decision logic** for 'normal' cases in decision tables.

As another example suppose that in determining *eligibility* of applicants for auto insurance:

- the standard considerations are: *driving history, evidence of insurance, insurance risk score, credit rating, and state/province.*
- an applicant must be denied auto insurance if either a *felon or under 18 years of age*.

Since *criminal status* and *age* respectively are not among the standard or typical considerations for the decision logic, these cases are deemed *exceptional cases*. Their outcome can be specified as follows:

Exception:

ineligible if the applicant

- has been convicted of a felony involving a motor vehicle, or
- is younger than 18 years of age.

Identifying exceptional cases, and treating them apart from the decision logic for 'normal' cases, can hugely simplify decision tables.

It also helps keep the number of 'normal' considerations as low as possible.

3.3 DEFAULTS

Omissions (missing rules) in a **decision table** can be addressed by a **default**.

Suppose the company with the **operational business decision**, *What should be charged for shipping an order?*, in Figure P2-1:

- ships orders to all *zip codes*, but
- has a standard price for shipping to many of them (say, *$39*).

Cases for all these same-price *zip codes* need not necessarily be listed in the decision table. Instead, a default can be specified to cover them all. A default can be expressed in different ways, including the two illustrated in Figure P3-3.

Observations

- The decision table on the left doesn't explicitly say 'default', but the top row in effect gives one.
- The decision table on the right – the preferred approach in **TableSpeak** – says 'default' explicitly.
- In both decision tables above, the **cases** covered by the default are given by the logical expression "*not 00401 and not 04401 and not 04402 …*".

  That kind of expression is *not* one you'd want to see in a decision table if you can possibly avoid it(!).

Figure P3-3. Two Ways of Asserting a Default

What should be charged for shipping an order?	
zip code	*shipping cost*
none of the zip codes listed below	$39
00401	$29
04401	$26
04402	$21
04730 to 04739	$35
04740	$19
~~	~~
~~	~~
99928 to 99929	$48
99950	$36

What should be charged for shipping an order?		
Default: *$39*	*zip code*	*shipping cost*
	00401	$29
	04401	$26
	04402	$21
	04730 to 04739	$35
	04740	$19
	~~	~~
	~~	~~
	99928 to 99929	$48
	99950	$36

Calling a default out explicitly as such (as in the decision table on the right) is the simplest, most business-friendly way to express it. Separating the default from the other rows and columns makes it much more visible – that way it doesn't get lost. (The default can also be included along with other semantics in a **decision box**.)

Notes

- Defaults are another important means by which decision tables can be simplified and kept as small as possible.
- A default in a decision table is simply another **decision rule**, one that supplies the appropriate outcome(s) for missing cases. A default can be used to ensure a decision table is **complete**.
- In software engineering a default might be viewed as *otherwise*, or as the *else* in *if-then-else*.
- A default and an **exception** are not the same thing. A default uses the 'normal' considerations for a decision table; an exceptional case *never* does. That way there is never any conflict about which one applies to any particular case.
- Defaults are by no means always as simple as a single quantity or instance. For example, a *set* of defaults might apply requiring a completely separate decision table.

Some approaches recommend avoiding defaults on the grounds they disturb the 'natural order' of well-structured decision tables. In real-life

business, however, people's *time* is expensive, so specifying defaults should be permitted.

Nonetheless:

> *Defaults should always be used very carefully.*

Defaults:

- Might leave important cases unexamined that should have distinct outcomes.
- Can easily be forgotten (but still apply) as decision logic evolves over time. Once (re)deployed, such un(re-)examined decision logic can produce unpleasant surprises.

3.4 RELEVANCE RESTRICTIONS

A **relevance restriction** always answers the following general **pattern question**:

> *Can an* **outcome** *from one significant* **operational business decision** *make any outcome from a second operational business decision meaningless?*

As applied to a **decision table** a relevance restriction always either:

- *Precludes* any outcome(s) for some **case(s)** from the second decision.

  In this role the relevance restriction produces a **preemption**.
- *Warns* that if any outcome *is* provided for certain case(s) from the second decision, the outcome is not necessarily valid.

  In this role the relevance restriction produces a **caveat**.

These roles are individually discussed and illustrated below.

3.4.1 Relevance Restrictions as Preemptions

DecisionSpeak recognizes **relevance dependencies** between **operational business decisions**.[12] A relevance dependency occurs when the answer to one question can completely eliminate the need or possibility for an answer to another question. In such cases the other operational business decision is *preempted* – made meaningless.

For example, suppose the business with the operational business decision *What should be charged for shipping an order?* in Figure P2-4 decides not to ship:

- *any* orders to certain *zip codes.* Possibly those cases are unprofitable, too slow, too dangerous, or too difficult to manage.
- *tubes* to certain *zip codes.* Possibly the postal facilities in those cases do not have the right kind of equipment.

For these particular cases:

- Asking the question *What should be charged for shipping an order?* is pointless because the answer to the more basic business question *Can an order be shipped to a customer?* is *no*.
- There is no *shipping charge* whatsoever (at least not one that's usable). Any *shipping charge* has been made meaningless – *preempted*.

Figure P3-4 shows the appropriate specifications for these restrictions.

The **decision cell(s)** for all preempted cases in the **decision table** have been:

[12] *DecisionSpeak Primer* - http://www.brsolutions.com/IPSpeakPrimers

Figure P3-4. Restriction for Preempted Cases by Zip Code and Kind of Packaging

What should be charged for shipping an order? Preemptions 1. 99928 thru 99929 2. tubes to 04402	kind of packaging	
zip code	box	tube
00401	$29	$35
04401	$26	$31
04402	$21	n/a
04730 to 04739	$35	$45
04740	$19	$19
~~	~~	~~
~~	~~	~~
99928 to 99929	n/a	n/a
99950	$36	$37

- **grayed-out**. Gray-out is a TableSpeak convention indicating no revision (updating) is permitted (at least until the restriction is lifted, if ever).
- specified as *n/a* (*not applicable*). Such a case is *preempted*; that is, no outcome for it is meaningful. Refer to Appendix 7 for additional discussion of *n/a*.

The advantage of retaining grayed-out decision cells with *n/a* in the decision table is to emphasize that the cases have not been overlooked. Omitting the cases from the decision table could be mistaken for **omissions** (missing rules), which they are *not*.

Observations

- The first preemption affects multiple cells (two). Defining the restriction only once allows **single-sourcing** of business intent.
- *Not applicable (n/a)* should be viewed as a definite outcome.

  The decision logic is basically saying this: "The case is recognized as being a **case in scope**, but the correct answer to the question being asked is that there *is* no answer to the question in this case."
- Preempted cases are not **scope items** – decision logic never gives definite outcomes for scope items per se.
- Preempted cases are not **exceptions** of any kind – preempted cases do not use **considerations** that are any different from 'normal' cases.
- Preemptions can be much more complex than those illustrated above. For example they might require specifying separate lists or tables. Refer to Part 5.7 for examples.
- Although the first restriction above affects multiple cells (two), its specification has been **single-sourced**.

  If some change(s) to the preemptions and/or to any of the affected outcome(s) are subsequently desired, the restrictions provide a single point to coordinate (permit) change to the outcome(s).

As an additional example, suppose the business with the operational business decision *What should be charged for shipping an order?* in Figure P2-2 decides not to ship orders that weigh 100 kgs or more. For these cases:

- Asking the question *What should be charged for shipping an order?* is pointless because the answer to the more basic business question *Can an order be shipped to a customer?* is *no*.
- There is no *shipping charge* whatsoever (at least not one you can use). Any *shipping charge* has intentionally been made meaningless – *preempted*.

Figure P3-5 shows the appropriate specifications for this restriction.

The **decision cell(s)** for preempted cases in this decision table (weight ≥ 100kgs) have not been shown. They don't have to be – that's an option.

If someone tried to update the decision table to show such a case, however, the restriction would have to be removed first. The original business intent has been retained and is being continuously enforced by the restriction.

3.4.2 Relevance Restrictions as Caveats

A relevance restriction that acts as a caveat is really a just a weaker version of a preemption. The decision logic basically says: "Here's an outcome, but don't count on it being valid because the business question hasn't been asked properly."

Consider the following example of relevance dependency between operational business decisions discussed in the **DecisionSpeak** Primer.[13]

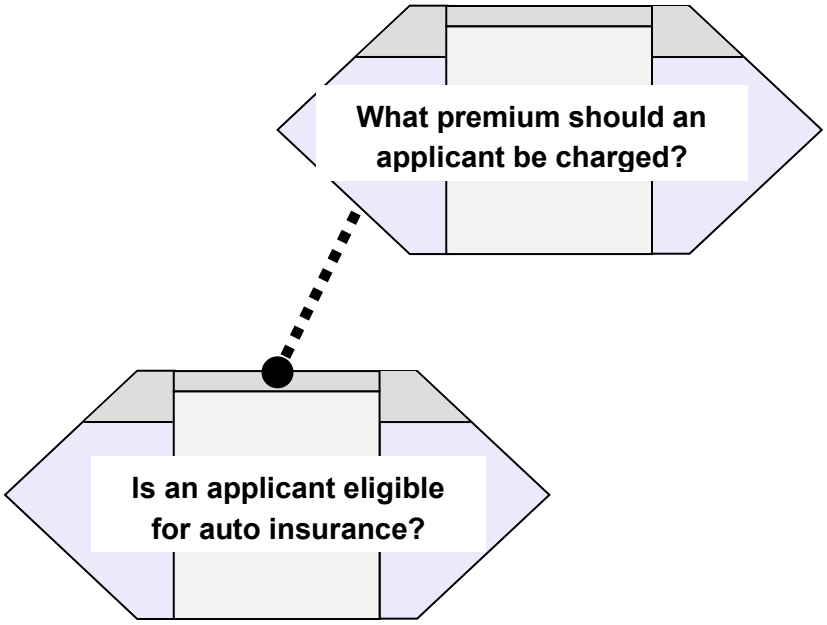
In determining eligibility of applicants for auto insurance, if an applicant is not eligible for coverage, there is no need to determine what to charge the applicant as a premium. This relevance dependency is illustrated in Figure P3-6 using a Q-Chart.

Figure P3-5. Restriction for Preempted Cases by Weight

What should be charged for shipping an order? Preemption 1. wt ≥ 100kg		weight (wt)						
		0kg ≤ wt < 1kg	1kg ≤ wt < 2kg	2kg ≤ wt < 3kg	3kg ≤ wt < 4kg	4kg ≤ wt < 5kg	~~	99kg ≤ wt < 100
zip code	00401	$29	$40	$81	$99	$171	~~	$1,276
	04401	$26	$36	$76	$99	$167	~~	$1,051
	04402	$35	$45	$83	$125	$145	~~	$927
	04730 to 04739	$35	$47	$73	$135	$135	~~	$836
	04740	$19	$25	$35	$62	$89	~~	$799
	~~	~~	~~	~~	~~	~~	~~	~~
	~~	~~	~~	~~	~~	~~	~~	~~
	99928 to 99929	$48	$72	$125	$155	$155	~~	$1,978
	99950	$36	$62	$125	$136	$136	~~	$1,592

[13] *DecisionSpeak Primer* - http://www.brsolutions.com/IPSpeakPrimers

Figure P3-6. Q-Chart Showing a Relevance Dependency Between Operational Business Decisions



Must processes always address the questions involved in a relevance dependency in bottom-to-top sequence? *No*.

For the questions in Figure P3-6, for example, a customer-friendly, web-based application might permit price-conscious consumers to ask about the premium *before* asking about eligibility.

Some precautions, however, should clearly be taken. Any outcome produced in a price-before-eligibility sequence is not necessarily valid.

The solution is to specify a restriction that requires a *caveat* to be issued along with the premium. The caveat, in the form of a message, is specified as part of the restriction. The message for the example above might be worded:

> *Securing coverage at the given price subject to eligibility.*

The restriction ensures a disclaimer is given by any process or use case that supports a price-before-eligibility sequence.

3.5 CONSIDERATION RESTRICTIONS

A **consideration restriction** always answers the following general **pattern question**:

> *Are there ways in which* **considerations** *should not be combined?*

A consideration restriction always constrains combination of **elemental cases** into **intersection cases** within a **decision table**. The net effect is to prohibit certain intersection cases that would otherwise be possible. Examples are provided in Figures P3-7 (only partially populated) and P3-8.

Figure P3-7. Example of a Consideration Restriction Prohibiting Combinations of Self with Self

What part can substitute for a recommended part in repairing equipment?	Consideration Restriction: *No part substituted for itself.* substitutable part					
recommended part	A2210	A5603	B9924	C7242	C4888	~~
A2210	n/a					
A5603		n/a				
B9924			n/a			
C7242				n/a		
C4888					n/a	
~~						

A part naturally cannot be substituted for itself. Such combinations (intersection cells) are meaningless given the question the decision table addresses.

In Figure P3-7 a consideration restriction has been specified (at top) to preclude such possibility.

Observations

- Although the restriction affects multiple cells, its specification has been **single-sourced**.
- **Gray-out** of the intersection cells indicates no revision (updating) is permitted (at least until the restriction is lifted, if ever).

For the example illustrated by Figure P3-8 let's assume that *traveling on business* (for the *location?* consideration) precludes *desire to cook?* (the consideration in the second column). The consideration restriction, *n/a if traveling on business,* has consequently been specified for the *desire to cook?* consideration.

The intersection cases given by the bottom three rows are thereby constrained – combinations including *yes* and *no* for the *desire to cook?* consideration are not permitted.

- *Yes* obviously needs to be disallowed because when traveling on business there is usually nowhere to cook.
- *No* is also disallowed because not *desiring* to cook is different than not being *able* to cook – it's not the same meaning.

Gray-out of intersection cells in Figure P3-8 indicates no revision (updating) is permitted for them (at least until the restriction is lifted, if ever).

Figure P3-8. Example of Elemental Case for One Consideration Precluding Any Elemental Case for Another Consideration

Where should a person eat breakfast?			
location?	*desire to cook? (n/a if traveling on business)*	*what clothes worn?*	*where to eat breakfast*
at home	yes	--	at home
at home	no	pajamas	at home
at home	no	jeans	casual restaurant
at home	no	dress	fancy restaurant
traveling on business	n/a	pajamas	room service
traveling on business	n/a	jeans	casual restaurant
traveling on business	n/a	dress	fancy restaurant

3.5.1 Use of Consideration Restrictions for Unknowns

A **consideration restriction** can be used to answer the special pattern question:

> *Can a* **decision** *still be made where does not matter (dash) applies and no* **elemental case** *is presented?*

Refer to Part 5.8 for discussion.

3.6 OUTCOME RESTRICTIONS

An **outcome restriction** always answers the following general **pattern question**:

> *Are certain* **cases** *limited to a particular* **outcome** *or to some subset of all* **potential outcomes***?*

An outcome restriction applied to a **decision table** directly constrains outcomes. The net effect of an outcome restriction is always either to:

- Establish a 'fixed' (or 'set') outcome for some **case(s)**.
- Narrow the set of potential outcomes for some case(s).

Use of outcome restrictions for each of these roles is discussed and illustrated individually below.

3.6.1 Fixing the Outcome for Some Case(s)

To illustrate a fixed outcome let's return to the operational business decision *What should be charged for shipping an order?* examined earlier.

Suppose the business wants to set a fixed price, *$250*, to ship orders to *zip code 99950* that weigh less than *5* kgs. Figure P3-9 illustrates how the fixed price can be specified as an outcome restriction.

Figure P3-9. Specifying a Fixed Outcome

What should be charged for shipping an order? Fixed Outcome 1. $250 for zip code 99950 & wt < 5 kg		weight (wt)						
		0kg ≤ wt < 1kg	1kg ≤ wt < 2kg	2kg ≤ wt < 3kg	3kg ≤ wt < 4kg	4kg ≤ wt < 5kg	~~	99kg ≤ wt < 100kg
zip code	00401	$29	$40	$81	$99	$171	~~	$1,276
	04401	$26	$36	$76	$99	$167	~~	$1,051
	04402	$35	$45	$83	$125	$145	~~	$927
	04730 to 04739	$35	$47	$73	$135	$135	~~	$836
	04740	$19	$25	$35	$62	$89	~~	$799
	~~	~~	~~	~~	~~	~~	~~	~~
	~~	~~	~~	~~	~~	~~	~~	~~
	99928 to 99929	$48	$72	$125	$155	$155	~~	$1,978
	99950	$250	$250	$250	$250	$250	~~	$1,592

Observations

- In Figure P3-9 the outcome restriction *$250 for zip code 99950 & weight < 5 kg* has been specified. All corresponding intersection cases in the decision table show the outcome *$250*.
- The outcome restriction affects multiple cells (five). Defining the restriction only once allows **single-sourcing** of business intent.
- Since the outcome restriction prohibits any of these outcomes from being changed, these intersection cells are **grayed-out**. Otherwise, direct assertions of prices differing from *$250* could easily occur.

3.6.2 Narrow the Set of Possible Outcomes for Some Case(s)

Rather than fixing outcomes, an outcome restriction can also be used to narrow the set of permissible outcomes for some or all cases. Figures P3-10, P3-11 and P3-12 illustrate.

Figure P3-10. Range of Permissible Outcomes

What should be charged for shipping an order? Permissible Outcomes 1. $19 ≤ shipping charge < $2,000		weight (wt)						
		0kg ≤ wt < 1kg	1kg ≤ wt < 2kg	2kg ≤ wt < 3kg	3kg ≤ wt < 4kg	4kg ≤ wt < 5kg	~~	99kg ≤ wt < 100kg
zip code	00401	$29	$40	$81	$99	$171	~~	$1,276
	04401	$26	$36	$76	$99	$167	~~	$1,051
	04402	$35	$45	$83	$125	$145	~~	$927
	04730 to 04739	$35	$47	$73	$135	$135	~~	$836
	04740	$19	$25	$35	$62	$89	~~	$799
	~~	~~	~~	~~	~~	~~	~~	~~
	~~	~~	~~	~~	~~	~~	~~	~~
	99928 to 99929	$48	$72	$125	$155	$155	~~	$1,978
	99950	$36	$62	$125	$136	$136	~~	$1,592

In Figure P3-10 the outcome restriction *$19 ≤ shipping charge < $2,000* is specified. All intersection cases in the decision table consequently must fall into this range. No modification (updating) of the table is allowed that would violate this restriction.

In Figure P3-11 the outcome restriction *same form of exercise on Monday, Wednesday, and Friday depending on precipitation* is specified. No modification (updating) of the decision table is allowed that would violate this restriction.

In Figure P3-12 the outcome restriction *cals on Sunday ≤ cals on Saturday* is specified. No modification (updating) of the decision table is allowed that would violate this restriction.

3.6.3 Special Uses of Outcome Restrictions

An outcome restriction can be used to answer the special pattern question:

Figure P3-11. Inter-Outcome Consistency

Which form of exercise should be done? Outcome Restriction: *same form of exercise on Monday, Wednesday, and Friday depending on precipitation*	*precipitation*	
day of week	rainy	not rainy
Monday	*treadmill*	*jogging*
Tuesday	*treadmill*	*jogging*
Wednesday	*treadmill*	*jogging*
Thursday	*treadmill*	*jogging*
Friday	*treadmill*	*jogging*
Saturday	*health club*	*golf*
Sunday	*health club*	*golf*

Figure P3-12. Inter-Outcome Dependency

What is the goal for exercise in terms of calories burned? Outcome Restriction: *cals on Sunday ≤ cals on Saturday*	*precipitation*	
day of week	rainy	not rainy
Monday	*150 cals*	*150 cals*
Tuesday	*200 cals*	*250 cals*
Wednesday	*300 cals*	*350 cals*
Thursday	*250 cals*	*250 cals*
Friday	*100 cals*	*100 cals*
Saturday	*500 cals*	*600 cals*
Sunday	*150 cals*	*150 cals*

Does a general rule still apply even if it disappears in expanding a decision table?

Refer to Part 5.10 for discussion of consolidation/expansion of decision tables. Refer to Part 5.11 for discussion of how outcome restrictions can preserve general rules in expansions.

An outcome restriction can also be used to clarify the meaning of empty cells in a decision table. Refer to the discussion of Figure P5-10 for an example.

ADVANCED TOPICS

PART 4: DESIGNING EFFECTIVE DECISION TABLES FOR BUSINESS

4.1 DECISION TABLES AND STRUCTURED BUSINESS VOCABULARY (CONCEPT MODELS)

No form of **business rule** expression or representation, including **decision tables**, is viable or complete if not based on a well-defined, well-structured business vocabulary. Knowing the business meaning of the words that appear in expressions or representations is a key element for success.

That you must know exactly what the words mean is so obvious it almost goes without saying. Especially in running a complex business (and what business *isn't* complex these days?!) the meaning of the words should never be taken as a 'given'.

Unfortunately, many approaches pay nothing more than lip service to this need. As a result, practitioners often end up having to learn the hard way.

> *Rapid* progress *in developing decision tables without ongoing coordination of vocabulary is usually just an illusion. The really hard part is in the words.*

A well-developed discipline exists for creating **structured business vocabularies** (**concept models**) backed by a powerful standard.[14] Our approach, based on that standard, is called **ConceptSpeak**[15].

A single, unified concept model is a prerequisite for creating any scalable, multi-use body of guidance, no matter how the business rules are represented or expressed.

Incidentally, ***true* completeness** of **decision logic** can be determined only by reviewing the underlying business vocabulary.

4.1.1 Illustration of Concept Model

How to build a complete concept model is beyond the scope of this discussion. Looking at the issue on a smaller scale, however, is quite instructive.

Figure P4-1 presents a draft graphical concept model supporting the *What coat should be worn?* **decision table** illustrated in Figure P1-1. The example follows ConceptSpeak conventions.

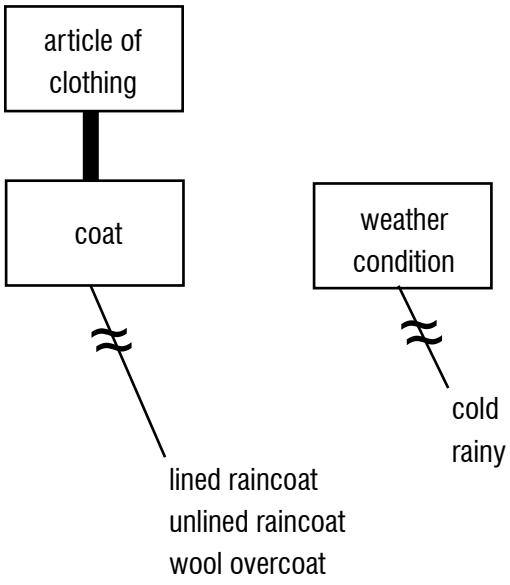
Observations

- Three general concepts are represented by the terms *article of clothing, coat,* and *weather condition.*
- *Coat* is a category of *article of clothing*.

[14] *Semantics of Business Vocabulary and Business Rules* (SBVR), released by Object Management Group (OMG) in late 2007. For an introduction and explanation, refer the *SBVR Insider* section on www.BRCommunity.com.

[15] **ConceptSpeak** is the BRS methodology for **structured business vocabularies** (**concept models**). Refer to *Business Rule Concepts* 4th ed (2013), Chapter 1 and Part 2. Available on http://www.brsolutions.com/b_concepts.php

Figure P4-1. Draft Graphical Concept Model for the *What coat should be worn?* Decision Table



- Three examples are indicated for *coat: lined raincoat, unlined raincoat,* and *wool overcoat*.
- Two examples are indicated for *weather condition: cold, rainy.*

Behind the graphical concept model are, of course, the all-important definitions (not shown).

Note that the concept model does not indicate any connection (verb concept) between *coat* and *weather condition* to represent the **decision rules** in the decision table. Doing so would create redundancy.

TableSpeak Best Practice

> *Decision rules per se should never be shown in concept models, just the basic vocabulary (noun and verb concepts) needed to express them.*

In general, a graphical concept model should represent only concepts that cannot be computed or derived by any business rule or found in some decision table. In other words, a graphical concept model should show only vocabulary that is structurally basic to the problem space.

4.1.2 Typical Change Scenario

Suppose the subject matter experts decide that what should be worn on a day that is *hot* is not the same as on a day that is *moderate*. In other words, just saying *not cold* is insufficient for the selectivity in **outcome** required.

Figure P4-2 presents **revised decision logic** (new **elemental cases** highlighted in purple).

Figure P4-2. Revised Decision Logic

		rainy?	
temperature level	**What coat should be worn?**	**rainy**	**not rainy**
	cold	*lined raincoat*	*wool overcoat*
	moderate	*unlined raincoat*	*sweater*
	warm	*umbrella*	*none*

Observations

- The **consideration** *Is it cold?* has been renamed *temperature level.* This consideration now shows three **elemental cases**: *cold, moderate,* and *warm*.
- There are now six intersection cells (3×2), rather than four (2×2). A total of six decision rules are now represented.
- The new outcomes *umbrella* and *sweater* have been introduced, as appropriate.

The original concept model is now badly out of sync with the decision table. Specifically, the original concept model does not include the new terms:

- *temperature level*
- *moderate* and *warm*
- *umbrella* and *sweater*

Worse is the following problem: Neither *umbrella* nor *sweater* is a *coat*! Even the wording of the question the **operational business decision** addresses is now awry. That problem must be fixed straightaway.

TableSpeak Best Practice

> *The question and the* outcomes *(answers) for each operational business decision should remain aligned at all times.*

Given the new decision rules the question addressed by the operational business decision might now more accurately be worded *What weather gear is appropriate?* A revised concept model is shown in Figure P4-3.

Observations

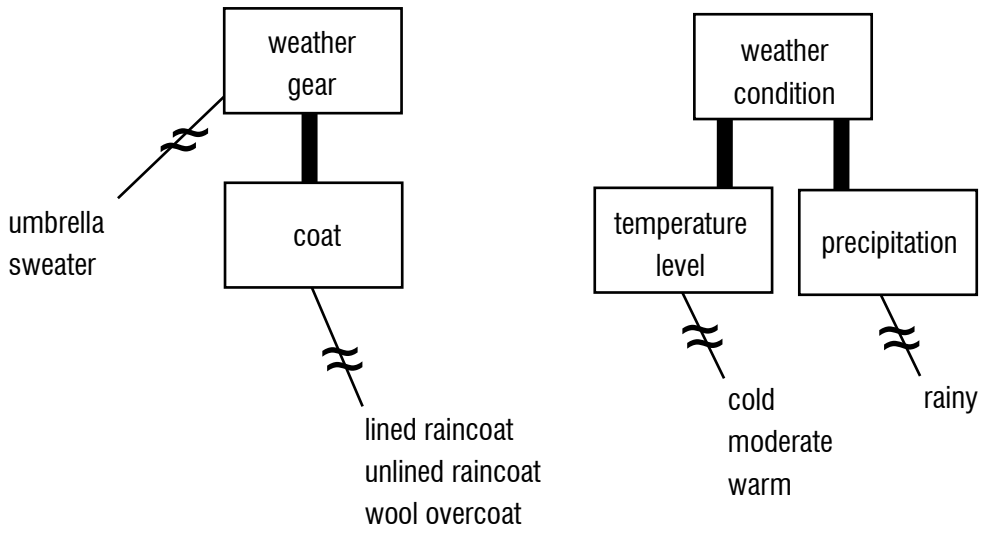
- The general concept *article of clothing* has been renamed *weather gear*.
- Examples of weather gear are: *umbrella* and *sweater*.
- *Temperature level* has been added as a category of *weather condition*.
- Examples of *temperature level* are: *cold, moderate,* and *warm*.
- *Rainy* is now shown as an example of *precipitation*, a new category of *weather condition.* (The addition of *precipitation* is presumably in anticipation of other examples besides *rainy.*)

4.1.3 Single-Sourcing Definitions and Names

Most styles of decision tables permit or require one or both of the following:

- The same elemental case to appear multiple times (often a great many times in the **row-or-column-style**).

Figure P4-3. Revised Concept model



- The same outcome to appear multiple times (in different decision rules).

Often the same elemental case or **potential outcome** also appears in *multiple* decision tables. With such extensive use, coordination is a significant concern – unfortunately one seldom highlighted by software vendors and many consultants.

Effective coordination of this business vocabulary requires at least two things, both provided by a concept model:

1. **Single-sourcing** of meaning (definitions).
2. Single point of change and traceability when any elemental case or potential outcome is re-named.

   On a *very* small scale such changes are illustrated by Figure P4-4.

Suppose the elemental case *pajamas* in the third column needs to be renamed *sleeping attire*. As illustrated in in Figure 4-4 (given earlier as Figure P3-8), changes need to be made in at least two places.

Important:

> *Never underestimate how frequently sets of elemental cases and potential outcomes might be revised*.

4.2 USING DECISION TABLES TO REPRESENT PROCEDURAL LOGIC

Problem Statement: A retail company has many departments, web-based lines of business, stores, etc. It tries to maintain the highest data quality possible for its customer master database. Updates are permitted only if the source is trusted.

Figure P4-4. Renaming an Elemental Case

Where should a person eat breakfast?			
location?	*desire to cook? (n/a if traveling on business)*	*what clothes worn?*	*where to eat breakfast*
at home	yes	--	at home
at home	no	~~pajamas~~ sleeping attire	at home
at home	no	jeans	casual restaurant
at home	no	dress	fancy restaurant
traveling on business	n/a	~~pajamas~~ sleeping attire	room service
traveling on business	n/a	jeans	casual restaurant
traveling on business	n/a	dress	fancy restaurant

Figure P4-5. Proposed Decision Table for the Customer Master Database Problem

	Should customer data be created or updated?		action
	new customer?		*create customer*
kind of customer	*existing customer?*	*line of business where this customer did business authorized to update?*	*update customer*
		line of business where this customer did business not authorized to update?	*none*

The following question might be proposed as the focal point for **decision analysis**: *Should customer data be created or updated?* Figure P4-5 presents a potential **decision table** to represent the related **decision logic**.

Problems with this representation include those identified in Table P4-1, which also presents related **DecisionSpeak** and **TableSpeak** best practices.

Table P4-1. Problems Evident in Figure P4-5 and the Associated TableSpeak Best Practices

Problem	Manifestation in Figure P4-5	Best Practice for Business-Oriented Decision Tables
actions	The outcomes shown in the rightmost column (highlighted in light blue) are actions. Their labels are clearly based on verbs in imperative (command) form.	Outcomes should never be something to do, but rather simply answers to some business question.
system behavior	The outcomes are not only actions, they are *system* actions (*create, update*).	Outcomes should never be something *system-ish* to do (e.g., create a new customer record). Never focus on capturing system logic for software development or control.
sequence	The order in which the considerations are evaluated matters. It makes no sense to ask whether the line of business where the customer did business is authorized to update *before* asking whether the customer is a new or an existing one.	A sequential dependency should never be embedded in a decision table.
empty or missing cell	The second row is missing a cell where an elemental case should go. Such omission is inevitable when a decision table embeds a sequential dependency between considerations.	Empty or missing cells (ones that should hold an elemental case) should be carefully avoided. They *always* signal some problem.

A **procedural** representation (e.g., a process model, use case, flowchart, etc.) is more natural and effective in capturing and representing logic where:

- The desired determination involves what action to take.
- The order (sequence) in which the factors are evaluated matters.

Logic trees can be effective for capturing and representing such logic. Figure P4-6 presents a logic tree to supplant the decision table above.

Observations

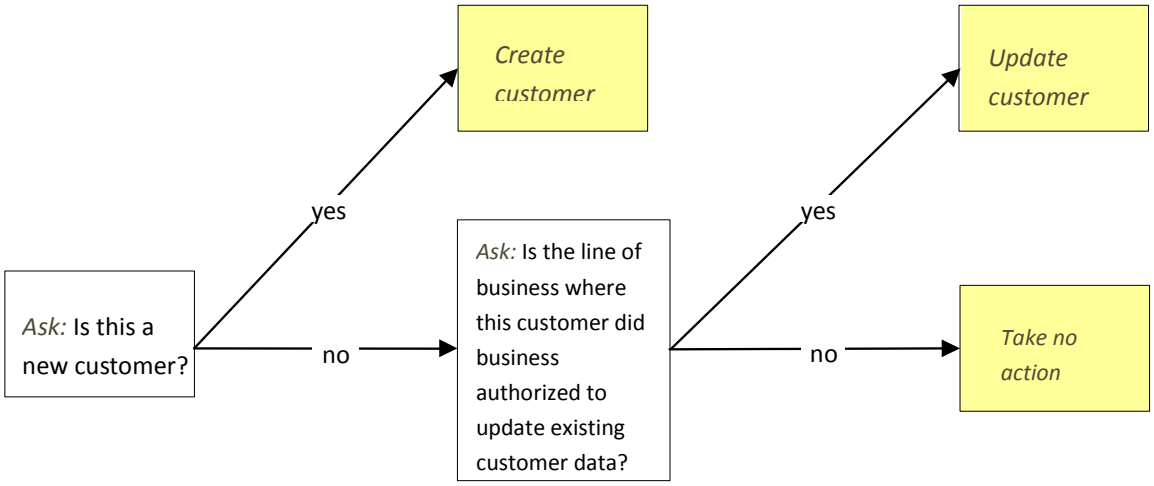
- The label for each box starts with a verb in the imperative (command) form (i.e., *ask, create, update, take*). Representations of procedural logic should always adhere to this naming convention for actions or tasks.
- The two *ask* boxes involve explicit questions. Asking questions in procedural representation of logic is fine with the following caveats:
    1. Such questions should always pertain to determining what to do next.
    2. If the order of asking the questions can be shown *not* to matter, a decision table should be used instead.

4.3 CONVERTING PROCEDURAL DEPENDENCIES INTO DECLARATIVE DEPENDENCIES

Problem Statement: The company needs to price its products appropriately. Some products are pre-owned (acquired from an external source), which makes a difference in its price. Prices should be determined as follows. If a product is:

- Pre-owned but not subsequently worked on in-house, its price should be the purchase price plus 12%.
- Pre-owned but significant work has been done on it, its price should be hours worked plus 15%.
- Standard, its list price should be charged.
- Custom, the contract price should be charged.

Figure P4-6. Logic Tree for the Customer Master Database Problem



The business question for this problem statement is *What should be charged for a product?* Figure P4-7 presents a potential (flawed) **decision table** to represent the **decision logic**.

Figure P4-7. Poorly-Designed Decision Table with a Sequential Dependency and Empty Cells

	What should be charged for a product?	worked-on?	price
kind of product	pre-owned	no	purchase price + 12%
		yes	hours + 15%
	standard		list price
	custom		contract price

Observations

- The decision table shows two **considerations**, *kind of product* and *worked on?.*
- *Price* is shown as the **outcome** (in the rightmost column).
- The second consideration, *worked on?,* applies only if the *kind of product* is *pre-owned*. As a consequence:
  1. Sequence is required in interpreting the decision table.
  2. Empty cells appear for the *worked on?* consideration.

The empty cells imply a second, more selective question that can be worded *What price should be charged for a pre-owned product?* This question should be addressed separately.

Figure P4-8 illustrates, eliminating the **sequential dependency** embedded in the original decision table.

Observations

- A decision table for a second, more selective **operational business decision**, *What should be charged for a pre-owned product?*, has been added at right. It has one consideration, *worked on?*.
- The outcome for this second decision table is *pre-owned product price* (in green[16]). This outcome also appears for elemental case *pre-owned* in the original, revised decision table (at left).
- The empty cells and sequential dependency have been eliminated from the original decision table. A **logical dependency** now exists between the two decision tables. This logical dependency is based on the common term, *pre-owned product price*.

In **business rules**, a logical dependency exists whenever one expression or representation uses some concept computed or derived by another.

Use of logical dependencies is a highly effective means to **single-source** different bits of business logic and to ensure reusability.

[16] **ConceptSpeak** recommends green to indicate that a concept is derived or computed.

Figure P4-8. Replacing a Sequential Dependency with a Logical Dependency

worked-on?	What should be charged for a pre-owned product?	pre-owned product price
	no	*purchase price + 12%*
	yes	*hours + 15%*

kind of product	What should be charged for a product?	price
	pre-owned	*pre-owned product price*
	standard	*list price*
	custom	*contract price*

PART 5: DESIGNING HIGH-QUALITY DECISION TABLES

5.1 TABLESPEAK RECOMMENDATIONS FOR SELECTING A REPRESENTATION STYLE

TableSpeak recommendations for selecting the best representation style for **decision tables** are presented in Table P5-1. The recommendations are based on the number of **few-case** and **many-case considerations**. The Table also color-codes and highlights the three fundamental style-related **TableSpeak** thresholds, which are explained in Table P5-2.

> *Note:* If table management software is available, you may be able to test-drive different representations for the same decision table to see which suits the business purpose best.

Legend for Table P5-1

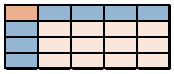
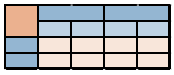
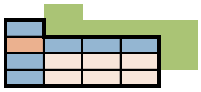
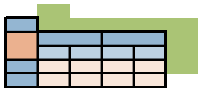
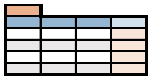
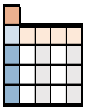
Icon	Meaning
	No intersection cases
	Intersection Style
	Intersection Style – with embedding
	Intersection Style – multi-table
	Intersection Style – multi-table with embedding
	One-Rule-Per-Row Style
	One-Rule-Per-Column Style
	The decision logic is too complex for a single decision table. Re-analyze and subdivide the operational business decision using a Q-Chart.

Table P5-1. TableSpeak Recommendations for Selecting a Representation Style

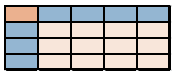
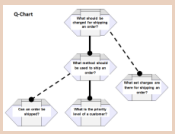
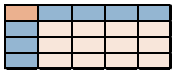
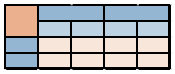
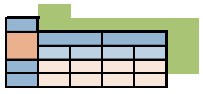
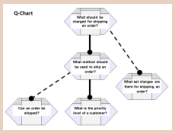
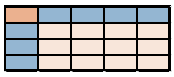
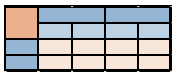
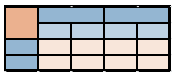
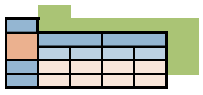
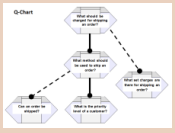
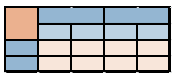
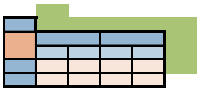
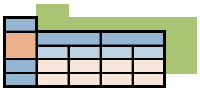
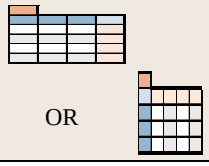
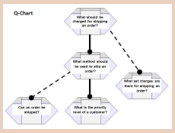
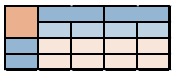
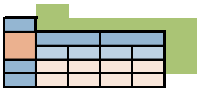
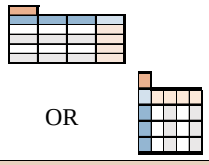
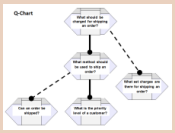
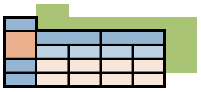
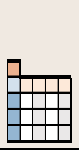
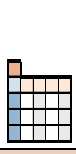
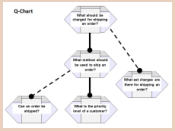
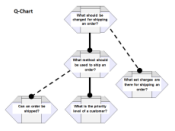
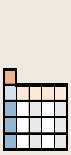
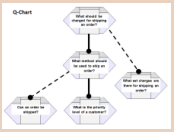
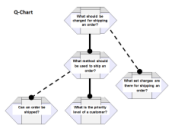
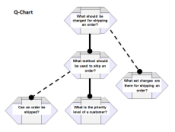
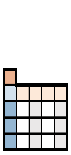
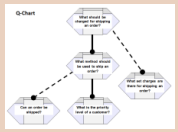
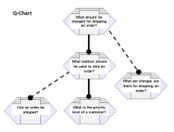
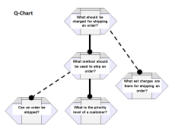
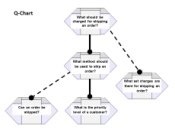
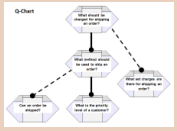
What is the best representation style for a decision table?						
		number of many-case considerations				
		0	1	2	3	3+
number of few-case considerations	0	n/a				
	1					
	2					
	3				OR 	
	4			OR 	OR 	
	5		OR 	OR 		
	6	OR 	OR 			
	7	OR 				
	7+					

Table P5-2. The Three Fundamental Style-Related Thresholds in TableSpeak

TableSpeak Threshold	Highlight Color in Table A5-1	Definition	Threshold	Resolution
multi-table threshold	light brown	the number of **many-case considerations** (three) at which multiple tables are apparently required in the **intersection style** to represent **decision logic**	3	Use a multi-table representation.
row-or-column threshold	light purple	the number of **considerations** at which the intersection style generally becomes impractical	6	Use the **row-or-column-style** instead.
complexity threshold	maroon	the number of considerations at which representation of the decision logic for an **operational business decision** using a decision table generally becomes impractical	The total number of many-case considerations exceeds 3, *or* the total number of all considerations exceeds 7.	Simplify or partition the operational business decision.

5.2 QUALITY ISSUES FOR ELEMENTAL CASES

Two important quality goals for **decision tables** are:

- **Completeness.** A decision table should give an **outcome** for every **case in scope**.
- **Certainty of outcome.** A decision table should never leave any doubt about what outcome is appropriate for any case in scope.

Improper or inadequate specification of **elemental cases** for a **consideration** can directly compromise both goals, as individually discussed and illustrated below.

5.2.1 Completeness

Figure P5-1 presents a revised version of Figure P1-1.

- In the original version the elemental cases for the consideration *Is it cold?* were: *Yes, it's cold* and *No, it's not cold*.
- In the revised version, the elemental cases have been changed to the temperature ranges *colder than 11°C* and *11°C ≤ temperature ≤ 27°C* (highlighted in purple).

Let's assume that the scope of the decision table was to cover all possible temperatures.

The revision fails to address the case *warmer than 27°C.* The decision table is compromised in that regard; it is *incomplete*.

Figure P5-1. Compromised Completeness Due to Missing Elemental Cases in Scope

		Is it rainy?	
		yes	**no**
Is it cold?	**colder than 11°C**	*lined raincoat*	*wool overcoat*
	11°C ≤ temperature ≤ 27°C	*unlined raincoat*	*none*

TableSpeak Best Practice

> *The elemental cases for any given consideration in a decision table should always be* **exhaustive**.

The same problem can arise in row-or-column-style decision tables. Figure P5-2, which gives the same **decision logic** as Figure P5-1, illustrates.

Figure P5-2. Compromised Completeness Due to Missing Elemental Cases in Scope in a Row-or-Column-Style Decision Table

Is it cold?	Is it rainy?	*What coat should be worn?*
colder than 11°C	**yes**	*lined raincoat*
	no	*wool overcoat*
11°C ≤ temperature ≤ 27°C	**yes**	*unlined raincoat*
	no	*none*

Again, what about *warmer than 27°C*? The elemental cases given for the *Is it cold?* consideration are not exhaustive. The decision table is compromised in that regard; it is *incomplete*.

5.2.2 Certainty of Outcome

Figure P5-3 presents another revised version of Figure P1-1. In this revised version, the elemental cases for the consideration *Is it cold?* have been changed to the temperature ranges *colder than 11°C* and *warmer than 9°C* (highlighted in purple).

Figure P5-3. Compromised Certainty of Outcome Due to Overlapping Elemental Cases

		Is it rainy?	
		yes	**no**
Is it cold?	**colder than 11°C**	*lined raincoat*	*wool overcoat*
	warmer than 9°C	*unlined raincoat*	*none*

Although the temperature ranges given in Figure P5-3 are now exhaustive, they overlap at 10°C. What are the appropriate outcomes for the intersection cases?

The decision table is compromised because the outcomes are uncertain.

- For *yes (it's rainy)* is the correct outcome *lined raincoat* or *unlined raincoat*? Either? Both?
- For *no (it's not rainy)* is the correct outcome *wool overcoat* or *none*?

  Note: These two outcomes are mutually-exclusive so the answer can't be 'both'.

TableSpeak Best Practice

> *The elemental cases for any given consideration in a decision table should always be* ***disjoint*** *(not overlapping).*

The same problem can arise in row-or-column-style decision tables. Figure P5-4, which gives the same decision logic as Figure P5-3, illustrates.

Figure P5-4. Compromised Certainty of Outcome Due to Overlapping Elemental Cases in a Row-or-Column-Style Decision Table

Is it cold?	Is it rainy?	*What coat should be worn?*
colder than 11°C	**yes**	*lined raincoat*
	no	*wool overcoat*
warmer than 9°C	**yes**	*unlined raincoat*
	no	*none*

Again, note that the temperature ranges given in Figure P5-4 overlap at 10°C. What are the appropriate outcomes for the intersection cases?

5.2.3 Brackets and Ranges as Elemental Cases

Elemental cases for a consideration can be specified as brackets or ranges. Examples:

- ranges of *temperatures*, as illustrated in the Figures above.
- ranges of *zip codes*, as illustrated in Figure P2-1.

How narrowly should brackets or ranges be specified?

TableSpeak Best Practice

> *If differential outcomes are likely to be needed for different cases within a bracket or range of a consideration, then the cases should be broken down.*

On the other hand, if the chances of needing differential outcomes in the future is small, then keeping a decision table as compact and as simple as possible easily trumps higher granularity.

5.3 EXPLICIT AND CLASS-BASED ORS IN ELEMENTAL CASES

Figure P5-5 includes **elemental cases** involving ORs for the **consideration** *day of week*. The OR is explicit in one elemental case and class-based in another.

Figure P5-5. An Example of Elemental Cases with ORs

	Which form of exercise should be done?	*precipitation*	
		rainy	not rainy
day of week	weekday	*treadmill*	*jogging*
	Sunday or Saturday	*health club*	*golf*

Observations

- An OR is explicit in the case *Sunday or Saturday.*
- An OR is implicit in the case *weekday. Weekday* as used here is a general concept (class) that actually stands for: *Monday* OR *Tuesday* OR *Wednesday* OR *Thursday* OR *Friday*.

A potential problem with this representation is that the appropriate **outcome** for any given *day of week* could change, independently and selectively. The **decision table** is not laid out to handle that possibility very well.

TableSpeak Best Practice

> *If differential outcomes are likely to be needed for ORed cases of a consideration, then the cases should be broken down.*

Note that this is the same best practice expressed in Part 5.2.3 above, except for brackets or ranges rather than ORed cases.

Based on the best practice, a better solution for the example above might be as presented in Figure P5-6.

Figure P5-6. Revised Decision Table Eliminating ORs in Elemental Cases

Which form of exercise should be done?	*precipitation*	
day of week	rainy	not rainy
Monday	*treadmill*	*jogging*
Tuesday	*treadmill*	*jogging*
Wednesday	*treadmill*	*jogging*
Thursday	*treadmill*	*jogging*
Friday	*treadmill*	*jogging*
Saturday	*health club*	*golf*
Sunday	*health club*	*golf*

Elimination of the ORs in Figure P5-6 has resulted in repetition of outcomes. For example, *treadmill* appears in five **decision cells**. Related observations:

- Decision tables are not required to show a different outcome for each **decision rule**. Each decision rule is entitled to its own outcome even if that outcome is the same as for other decision rules.
- Suppose the outcome *treadmill* needed to be changed in every decision rule, say to *aerobic dance*. A global replace would suffice to make the change.

  Note: Changing multiple outcomes is not the same as *renaming* the underlying **potential outcome**. For the latter, the best approach is coordination through a **structured business vocabulary** (**concept model**). Refer to Part 4.1.3.
- If each of the decision rules is *required* to have the same outcome, an **outcome restriction** should be specified. Refer to Part 3.6.

Common-Outcome Symbol. In larger decision tables, repetition of a rather long outcome can be distracting.

If desired, a placeholder symbol, say an X, can stand-in for an oft-repeated outcome. That way *non*-X outcomes show up much better.

The meaning of the common-outcome symbol (i.e., what outcome it represents) can be specified either in a **decision box** or within the decision table itself.

Contraction/Expansion. Automated support is desirable for optionally contracting/expanding larger decision tables based on ORs.

For example, such support might automatically convert the decision table in Figure P5-6 back into the one in Figure P5-5. The resulting *contraction* can assist in detecting overall patterns, especially **general rules**.

Refer to Part 5.9 for discussion of contraction/expansion for decision tables based on **subsumption**. Addressing ORs *within* elemental cases as in Figures P5-5 and P5-6 is simply a special case of subsumption.

5.4 TABLESPEAK GUIDELINES FOR COMPOSING OUTCOMES

In the majority of **decision tables**, **outcomes** are simply quantities, names, or individual words

(e.g., *yes, no*). Outcomes can be more complex, however. For example, outcomes can be some:

- arithmetic expression or formula
- phrase or sentence in natural language
- name for another decision table
- icon (For an example refer to Table P5-1.)

If an outcome is the name of a computed or derived concept, then some other decision table or **business rule** should indicate how to compute or derive it.

For example, Figure P4-7 shows *pre-owned product price* being computed by a second decision table. Such connections reflect **logical dependencies**.

5.4.1 Outcomes Embedding ORs

Although ORs among outcomes is a tricky issue in logic, they are perfectly acceptable in business. For example, the outcome *jogging or soccer* might be specified as a suitable *form of exercise,* say on *Thursday* when it's *not rainy*.

Such specification, incidentally, would *not* indicate that only one form of exercise per day is permitted. Any such limitation should be stated separately as an explicit **outcome restriction**.

5.4.2 Outcomes Embedding ANDs

Outcomes composed in a way that embeds some AND(s) raise special concerns. Among these concerns is reduced responsiveness to change. Figure P5-7 illustrates.

Figure P5-7. An Example of Outcomes with ANDs

	Which form of exercise should be done?	*precipitation*	
		rainy	not rainy
day of week	Monday	*treadmill, 150*	*jogging, 150*
	Tuesday	*treadmill, 200*	*jogging, 250*
	Wednesday	*treadmill, 300*	*jogging, 350*
	Thursday	*treadmill, 250*	*jogging, 250*
	Friday	*treadmill, 100*	*jogging, 100*
	Saturday	*health club, 500*	*golf, 600*
	Sunday	*health club, 400*	*golf, 400*

Observations

- In this revision of Figure P5-6 a new number has been inserted alongside each outcome. What that number represents is currently a mystery. More about this mystery number momentarily.
- The commas in each cell presumably represent ANDs. The layout of the decision table would therefore seem to indicate that for any given **intersection case** both the *form of exercise* and the mystery number apply.

In general, any outcome in a decision table that embeds an AND should be examined carefully. (An exception is if the AND is naturally part of some name – for example, *hide and seek*.)

At issue is the possibility of change. For example, for the decision *Which form of exercise should be done?* suppose that:

- It becomes necessary or desirable to consider whether an exercise companion is involved – i.e., to add *exercise companion* as a **consideration**.
- The mystery number in Figure P5-7 remains independent of this new consideration.

How should the revised **decision logic** be accommodated? The underlying problem with the decision table in Figure P5-7 is that the question the decision logic answers is no longer aligned with the outcomes. Clearly numbers are not any *form of exercise*.

TableSpeak Best Practice

> *Fundamental alignment between question and outcomes in a decision table should be maintained rigorously at all times.*

If maintaining alignment between questions and outcomes in a decision table proves problematic, return to higher-level **decision analysis** (**DecisionSpeak**).

To continue the example, suppose the mystery number in Figure P5-7 proves to represent the goal for exercising in terms of *calories burned*.

This outcome reflects a different **operational business decision** than selecting the appropriate *form of exercise*. The two operational business decisions need to be distinguished. **TableSpeak** puts it this way:

> *Different kinds of outcome – different questions, different decisions.*

Figure P5-8 provides appropriate revisions to the decision logic presented in Figure P5-7. Two distinct decision tables are evident, each with an appropriate question.

The revision also includes the new consideration *exercise companion* for determining the *form of exercise* (incomplete).

Figure P5-8. Separate Decisions for Distinct Kinds of Outcomes

Which form of exercise should be done?	*exercise companion(s)? / precipitation*			
	no companion		with companion(s)	
day of week	rainy	not rainy	rainy	not rainy
~~	~~	~~	~~	~~
Saturday	*health club*	*rowing*	*handball*	*golf*
Sunday	*indoor swimming*	*bicycling*	*indoor tennis*	*golf*

What is the goal for exercise in terms of calories burned?	*precipitation*	
day of week	rainy	not rainy
Monday	*150 cals*	*150 cals*
Tuesday	*200 cals*	*250 cals*
Wednesday	*300 cals*	*350 cals*
Thursday	*250 cals*	*250 cals*
Friday	*100 cals*	*100 cals*
Saturday	*500 cals*	*600 cals*
Sunday	*400 cals*	*400 cals*

5.4.3 General Concepts as Outcomes

Most operational business decisions are about selecting *specific* outcomes that are either:

- Some individual concept (e.g., *John Smith*).
- Some computed or derived concept (e.g., *total price*).

Showing a general concept as an outcome (e.g., *country*) should be analyzed carefully. In effect, the outcome represents an implicit AND among all instances of the concept. Is that result really intended? Doubtful.

5.5 REDUNDANCIES, MULTIPLE OUTCOMES, AND CONFLICTS

Anomalies among **business rules** are a well-known issue.[17] By the way, **declarative** representation of business rules does not somehow *cause* anomalies. Just the opposite, it makes anomalies far easier to detect compared to **procedural** representation.

Three important kinds of anomalies in **decision tables** are discussed in Table P5-3, which also examines implications for **intersection-style** and **row-or-column-style** decision tables. These anomalies all arise from lack of uniqueness for

Table P5-3. Potential Anomalies in Decision Tables Arising from Non-Unique Intersection Cases

Kind of Anomaly	Description of Anomaly	The Anomaly in Intersection Style Decision Tables	The Anomaly in Row-or-Column-Style Decision Tables
special note		*Uniqueness of* **intersection cases** *is a product of the physical structure of the matrix. There is always one and only one* **decision cell** *for each intersection case. Anomalies of the kinds listed below are either unlikely to occur or easy to spot.*	*Uniqueness of intersection cases must be strictly enforced (usually by software). Otherwise, in decision tables of any size, rows representing the very same intersection cases may prove hard to spot and the kinds of anomalies listed below can easily occur. (The same applies for columns in the one-rule-per-column style.)*
redundancy	the same **decision rule** appears more than once	Duplication of any specific intersection case, and therefore of the associated decision rule, is highly unlikely.	A new row can be added that addresses the same intersection case as an existing row. If the outcomes of the existing and new rows are also the same, the decision rules are exact duplicates.
multiple outcomes	the same case has two or more **outcomes**	A cell with more than one outcome is highly visible.	A new row can be added that addresses the same intersection case as an existing row. If the outcomes of the existing and new rows are not the same, the same intersection case now has multiple outcomes.
conflict	two or more decision rules share the same **consideration(s)** and cover exactly the same cases, but have mutually-exclusive outcomes	A cell with more than one outcome is highly visible. If the outcomes asserted there are mutually exclusive, a conflict arises. High visibility allows the conflict to be readily detected – or better yet, avoided in the first place.	A new row can be added that addresses the same intersection case as an existing row. If the outcomes of the existing and the new rows are mutually exclusive, a conflict arises.

intersection cases (i.e. a decision table not being **single-hit**).

5.5.1 More about Multiple Outcomes

Operational business decisions examined in **decision analysis** under **DecisionSpeak** and **TableSpeak** usually have one and only one outcome for each relevant case.

Multiple outcomes for a case can cause confusion about what the correct outcome really is – even if the outcomes are not mutually exclusive (i.e., not in conflict). In other words multiple outcomes can reduce **certainty of outcome** and also can result in inconsistent decisions. Cases with multiple outcomes should be examined carefully.

- If the outcomes are of different kinds, more than one question (operational business decision) is probably involved. Those questions should be analyzed separately.
- If an OR is intended (i.e., either outcome is acceptable), the OR should be expressed explicitly.
- An intentional AND between multiple outcomes of the same kind are rare for problems addressed by decision analysis. Perhaps a consideration is missing for one of the outcomes.

5.5.2 Resolving Conflicts

The prevention, detection and elimination of conflicts are extremely important requirements for operational business decisions addressed by decision analysis. If undetected, conflicts can lead directly to inconsistent decisions. In general, only business people or business analysts can resolve conflicts.

5.6 OMISSIONS – MISSING DECISION RULES

Omissions (missing decision rules) are an **anomaly** of significant concern in **decision analysis**. A **decision table** that lacks some **decision rule(s)** is not complete. **Certainty of outcome** is compromised. *The* **decision logic** *has holes.*

Omissions in decision tables can occur for any of the reasons enumerated in Table P5-4. The table also discusses:

- appropriate remedies.
- implications for **intersection-style** and **row-or-column-style** decision tables.

In addition to the remedies suggested in Table P5-4, other important ways in which omissions in a decision table can be addressed are to specify some:

- general rule(s).
- exception(s). Refer to Part 3.2 for discussion.
- default(s). Refer to Part 3.3 for discussion.

As indicated in Table P5-4, one reason for omissions is that a decision table is not **exhaustive** with respect to **intersection cases**.

Remember that being exhaustive is not the same thing as being non-redundant. A decision table can exhibit no **redundancy** with respect to intersection cases without being exhaustive – and vice versa. Refer to Part 5.5 for additional discussion.

Table P5-4 Omissions in Decision Tables and Remedies

Reason for Omission	Implication	Remedy	Comment for Intersection-Style Decision Tables	Comment for Row-or-Column-Style Decision Tables
An **elemental case** (within scope) does not appear.	The decision table is not exhaustive with respect to *elemental cases.*	1. Add all omitted elemental cases. 2. Ensure they are properly represented in every appropriate intersection case. 3. Indicate the appropriate outcome for each new intersection case.	For decision tables of any size, an exhaustive set of elemental cases for each consideration needs to be guaranteed by software.	*(same as for intersection style)*
An **intersection case** does not appear.	The decision table is not exhaustive with respect to *intersection cases.*	1. Add all omitted intersection cases. 2. Indicate the appropriate outcome for each new intersection case.	An exhaustive set of intersection cases is a product of the physical structure of the matrix. There is always a decision cell for each intersection case.	An exhaustive set of intersection cases needs to be guaranteed by software. Otherwise, in decision tables of any size, missing rows can easily occur and are hard to spot. (The same applies for columns in the one-rule-per-column style.)
A **decision cell** is empty.	The decision table is not exhaustive with respect to *decision rules.*	Indicate the appropriate outcome.	A decision cell with no outcome is highly visible. The matrix has an empty spot.	A row has an empty spot, which should be apparent upon inspection.

TableSpeak Best Practice

> *Ensure that decision tables are both exhaustive and non-redundant.*

Completeness in decision tables (in any style) is based on exhaustively addressing all elemental cases and all intersection cases. It is *not* based on simply addressing all **potential outcomes** at least once.

For this reason **TableSpeak** generally does not recommend organizing decision tables around common or similar **outcomes** (except perhaps during initial capture). Lexicographical order is preferred.

TableSpeak Best Practice

> *Do nothing in the format of a decision table that will detract from achieving completeness with respect to elemental and intersection cases.*

5.6.1 Sparse Style of Decision Tables

Decision tables with a small number of **potential outcomes**, often just two, occasionally feature a significant preponderance of some **outcome(s)** in comparison to some other outcomes(s). For example, *yes* as an outcome might be 10 or 50 or a 100 times more common than *no*.

To visually highlight some rare or unusual outcome(s) such decision tables are sometimes presented in **sparse style**.

In sparse style all intersection cases are deliberately omitted except for those with the rare or unusual outcome(s). The usual prescription against missing decision rules is relaxed. Refer to discussion of Figure P5-10 for an example.

So that there are no gaps in decision logic, **TableSpeak** recommends that the sparse style be used only where one of the following is true:

- The decision table is specified as a **preemption table** (as in Part 5.7).
- Some **default(s)** and/or **outcome restriction** for the common or usual outcome(s) are specified. (An outcome restriction is specified in Figure P5-10.)

5.7 PREEMPTION LISTS AND PREEMPTION TABLES

Preemptions can be complex for any of several reasons:

- A logical expression of arbitrary complexity is specified to identify relevant **cases**.
- A subset of relevant cases for a **many-case consideration** is significant in size.

The latter often requires specification of one or more *preemption lists* or *preemption tables*. Each kind is discussed and illustrated individually below.

5.7.1 Preemption Lists

A **preemption list** is simply *a special form of* **relevance restriction** *in which a long list of cases is required (typically four or more)*. Figure P5-9 provides examples.

Figure P5-9 includes three preemption lists. (Since there are tens of thousands of *zip codes*, the three preemption lists are presumably not at all typical in size.) The lists respectively indicate that the business chooses not to ship orders in the following cases:

- Any order to *zip codes* 04401 or 04740.
- Any order involving a *tube* to *zip codes* 00410 or 04730 to 04739
- Any order involving a *box* in the *summer* to *zip codes* 99928 to 99929 or 99950.

Observations

- The three preemption lists are specified separately from the **decision table** and referenced by name.
- Each list specifies cases (*zip codes*) to which orders are not to be shipped according to qualifications included in the name of the list.

  For example, the qualifications for the preemption list at the upper right are *boxes* and *summer*.
- All **intersection cases** in the main decision table corresponding to entries in the preemption lists show the **outcome** *n/a* (*not applicable*).

  If an order falls into a cell indicating it cannot be shipped, a price for shipping that order obviously should not be given. Such a case is *preempted*; that is, no outcome for it is meaningful. (Refer to Appendix 7 for additional discussion of *n/a*.)

Figure P5-9. Preemptions Using Three Preemption Lists

zip codes to which orders not sent
~~
04401
~~
04740
~~

zip codes to which tubes not sent
~~
00401
~~
04730 to 04739
~~

zip codes to which boxes not sent in summer
~~
~~
~~
99928 to 99929
99950

What should be charged for shipping an order?

Preemption Lists
1. *zip codes to which orders not sent*
2. *zip codes to which tubes not sent*
3. *zip codes to which boxes not sent in summer*

		weight and kind of packaging									
		0kg ≤ wt < 1kg		*1kg ≤ wt < 2kg*		*2kg ≤ wt < 3kg*		*3kg ≤ wt < 4kg*		*4kg ≤ wt < 5kg*	
zip code	*season*	*box*	*tube*	*box*	*tube*	*box*	*tube*	*box*	*tube*	*box*	*tube*
00401	summer	$29	n/a	$40	n/a	$81	n/a	$99	n/a	$171	n/a
	not summer	$41	n/a	$55	n/a	$101	n/a	$131	n/a	$203	n/a
04401	summer	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	not summer	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
04730 to 04739	summer	$35	n/a	$47	n/a	$73	n/a	$135	n/a	$135	n/a
	not summer	$25	n/a	$49	n/a	$73	n/a	$131	n/a	$135	n/a
04740	summer	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	not summer	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
~~	~~	~~	~~	~~	~~	~~	~~	~~	~~	~~	~~
99928 to 99929	summer	n/a	$37	n/a	$49	n/a	$101	n/a	$101	n/a	$109
	not summer	$115	$115	$125	$125	$200	$200	$200	$200	$250	$250
99950	summer	n/a	$37	n/a	$48	n/a	$98	n/a	$99	n/a	$110
	not summer	$115	$115	$125	$130	$219	$229	$251	$300	$303	$309

- All preempted cells have also been **grayed-out**.

  Gray-out indicates no direct revision (updating) is permitted (at least until the preemption is lifted, if ever).

  For convenience, different colors of shading have been applied for the gray-out to show which intersection cases have been preempted by which preemption list.

- Each entry in each preemption list results in multiple intersection cases being preempted.

- Tying multiple preempted cases to a single entry in this fashion allows **single-sourcing** of business intent.

  If revision is desired to a preemption list (either to drop some zip code(s) or to add some), the preemption list provides a single point of change.

- If a preemption list would be significantly shorter were it to include *zip codes* to which orders *are* sent (an *inclusion list*), rather than ones to which they are *not* sent (an *exclusion list*), the former may be used instead.

  Which kind of list is best for any given situation is simply a matter of preference (and proper specification).

5.7.2 Preemption Tables

Preemptions can sometimes be highly specific or selective. Specifying each preemption individually can be cumbersome and wordy. A special *preemption table* may substitute.

A **preemption table** is *a decision table set up specifically to preempt cases in another decision table.*

Naturally, the considerations for the respective decision tables must be parallel.

As an example, suppose the business does not ship orders where:

- The zip code is from 99928 to 99929, the *weight* is 0kg ≤ wt < 1kg, the *kind of packaging* is *tube*, and the *season* is *not summer*.
- The *zip code* is 00401, the *weight* is 3kg ≤ wt < 4kg, the *kind of packaging* is *box*, and the *season* is *summer*.
- The *zip code* is from 99928 to 99929, the *weight* is 4kg ≤ wt < 5kg, the *kind of packaging* is *tube*, and the *season* is *summer*.

These specifications could be individually listed as preemptions in the decision table in Figure P5-9.

As illustrated in Figure P5-10, however, a more convenient approach is to create a separate preemption table and to simply reference it by name from the former decision table.

Figure P5-10. Preemption Table

Preemption Table *orders that cannot be shipped (empty cell → can be shipped)*		*weight and kind of packaging*					
		0kg ≤ wt < 1kg		3kg ≤ wt < 4kg		4kg ≤ wt < 5kg	
zip code	*season*	*box*	*tube*	*box*	*tube*	*box*	*tube*
00401	summer			no			
	not summer						
99928 to 99929	summer						no
	not summer		no				

Observations

- Figure P5-10 is organized along the same lines as the main decision table in Figure P5-9 – that is, in the same style and with the same considerations.
- This decision table indicates whether an order with certain characteristics can be shipped at all. A *no* indicates it cannot.
- Since most cases *can* be shipped the decision table is shown in **sparse style**.

  In sparse style the usual prescription against missing intersection cases is relaxed. Only intersection cases with a specific outcome (e.g., a *no*) are represented so that those specific cases really stand out.
- An **outcome restriction** has been specified to indicate that empty cells are to be interpreted as *can be shipped*.
- Each of the three corresponding intersection cases in the original decision table should now show *n/a* (*not applicable*) and be *grayed-out*.

5.8 USE OF CONSIDERATION RESTRICTIONS AGAINST UNKNOWNS

A **consideration restriction** can be used to answer the special **pattern question**:

> *Can a* **decision** *still be made where does not matter (dash) applies and no* **elemental case** *is presented?*

Refer to Appendix 7 for discussion of *does not matter* (dash) as used in **TableSpeak**. Use of the symbol raises the issue of how to handle *unknowns.* This issue can arise only when this special symbol appears.

Typically, **decision logic** requires elemental cases for each consideration of an **intersection case** to be present (known). If not, the appropriate **outcome** cannot be selected and no decision can be made.

Specifying *does not matter* (dash), however, raises the following possibility. Since no elemental case for the consideration can possibly make a difference to the outcome, perhaps the *absence of any* can be tolerated.

Consider the **operational business decision** *What is the right delivery method for an order?.* Suppose *does not matter* (dash) is specified in some intersection case for the *hazardous materials* consideration. Which of the following is meant?

1. There is no difference in outcome for any **elemental case** that can be presented – *and* that outcome still applies even if no elemental case at all is presented.
2. There is no difference in outcome for any elemental case that can be presented – *but some* elemental case must nonetheless still be presented.

The latter version is the more restrictive so it requires additional specification. When absence is *not* tolerable the consideration restriction *must be known* should be specified.

As a result, *some* elemental case must present itself for a decision to be made – even though the specific elemental case is inconsequential to the outcome.

Figure P5-11 illustrates specification of *must be known* for the *hazardous materials* consideration (purple cell).

Figure P5-11. Consideration Restriction Indicating *Must Be Known*

What is the right delivery method for an order?							
rush order?	*any fragile item?*	*any specialty item?*	*any high-priced item?*	*any item involving hazardous materials? (must be known)*	*category of customer?*	*destination of order?*	***delivery method for order***
no	*no*	*no*	*no*	*no*	*silver*	—	**picked up by customer**
yes	*yes*	*no*	*no*	*yes*	*gold*	*local*	**shipped by normal service**
yes	—	—	—	*yes*	*platinum*	*remote*	**shipped by premium service**

A particular intersection case might not start off with a *does not matter* (dash), but rather might acquire it later on as revisions to the decision table are made.

- In Figure P5-11, for example, the intersection case in the last row starts off *yes* for the *hazardous materials* consideration (see light-purple cell).
- In Figure P5-12 that specification has changed to *does not matter* (dash).

The consideration restriction still requires a *yes* or *no* to be presented for any decision to be made.

Specifying *must be known* for a consideration is not the same as specifying some property (or field or attribute) as mandatory:

- A mandatory property must be known *all the time*.
- A *must be known* consideration applies only at the specific time any decision is made.

If a property must be known *before or after* some decision is made, some other **business**

Figure P5-12. Intersection Case Includes Does Not Matter (Dash)

What is the right delivery method for an order?							
rush order?	*any fragile item?*	*any specialty item?*	*any high-priced item?*	*any item involving hazardous materials? (must be known)*	*category of customer?*	*destination of order?*	***delivery method for order***
no	*no*	*no*	*no*	*no*	*silver*	—	**picked up by customer**
yes	*yes*	*no*	*no*	*yes*	*gold*	*local*	**shipped by normal service**
yes	—	—	—	—	*platinum*	*remote*	**shipped by premium service**

rule should be specified.

A consideration restriction specifying *must be known* may be applied in **decision tables** either to:

- Some consideration as a whole (as in Figure P5-11 and P5-12). Such specification applies individually to every intersection case.
- Some particular intersection case.

5.9 SUBSUMPTION IN DECISION TABLES

Subsumption simply means that one thing is viewed or encompassed as a member of a larger class of things.

In **business rules**, **subsumption** occurs when a **general rule** is recognized to cover exactly the same situation, and to give or require exactly the same result, as a more specific rule. An undetected subsumption (that is, a rule that could be subsumed) is always considered an **anomaly**.

The matter is complicated a bit in **decision tables**.

Deliberate *expansion* of decision tables – in effect reversing subsumptions (usually by software) – can be a useful technique for visualization and validation. Contraction/ expansion of decision tables is discussed in Part 5.10.

5.9.1 Examples of Subsumption in Decision Tables

The decision table in Figure P5-13 illustrates potential subsumption. A new **decision rule** (the new row in light purple at bottom) has been added to the decision table in Figure P2-7.

This new decision rule is exactly like the one in the row immediately above it in two respects. They both:

- Pertain to *platinum* customers.
- Have the same **outcome**, *shipped by premium service*.

The new decision rule, however, is more general since it includes a *does not matter* (dash) for *all* the other **considerations**.

Since this new decision rule covers a broader set of **intersection cases**, but leads to the same outcome, the decision rule in the row above it is

Figure P5-13. Decision Table with a Subsumable Decision Rule

What is the right delivery method for an order?							
rush order?	*any fragile item?*	*any specialty item?*	*any high-priced item?*	*any item involving hazardous materials?*	*category of customer?*	*destination of order?*	***delivery method for order***
no	*no*	*no*	*no*	*no*	*silver*	—	**picked up by customer**
yes	*yes*	*no*	*no*	*yes*	*gold*	*local*	**shipped by normal service**
yes	—	—	—	*yes*	*platinum*	*remote*	**shipped by premium service**
—	—	—	—	—	*platinum*	—	**shipped by premium service**

Figure P5-14. Contracted Decision Table Not Showing Subsumed Decision Rule

What is the right delivery method for an order?							
rush order?	*any fragile item?*	*any specialty item?*	*any high-priced item?*	*any item involving hazardous materials?*	*category of customer?*	*destination of order?*	***delivery method for order***
no	*no*	*no*	*no*	*no*	*silver*	—	**picked up by customer**
yes	*yes*	*no*	*no*	*yes*	*gold*	*local*	**shipped by normal service**
—	—	—	—	—	*platinum*	—	**shipped by premium service**

Figure P5-15. Decision Table with Subsumable Decision Rules

What is the right delivery method for an order?							
rush order?	*any fragile item?*	*any specialty item?*	*any high-priced item?*	*any item involving hazardous materials?*	*category of customer?*	*destination of order?*	***delivery method for order***
no	*no*	*no*	*no*	*no*	*silver*	—	**picked up by customer**
yes	*yes*	*no*	*no*	*yes*	*gold*	*local*	**shipped by normal service**
yes	—	—	—	*yes*	*platinum*	*remote*	**shipped by premium service**
no	*no*	*no*	*no*	*yes*	*silver*	—	**picked up by customer**

now subsumable. That row could be absorbed within a contracted decision table as illustrated in Figure P5-14.

Subsumptions are usually not so easy to detect as in the example above.

Figure P5-15 presents a somewhat more complicated example, also based on the original decision table in Figure P2-7. Again, a new decision rule (the new row in light purple at bottom) has been added.

This new decision rule is exactly like the one shown in the top row in two respects:

- They both cover exactly all the same **elemental cases** – except for one, the elemental case pertaining to the consideration *hazardous materials.*
- They both have the same **outcome**, *picked up by customer*.

With regard to that one difference pertaining to *hazardous materials*, the new row shows the elemental case *yes*, whereas the top row shows the elemental case *no.* Since both *yes* and *no* lead to the same outcome, these two decision rules are subsumable into a more general rule (row).

Figure P5-16. Contracted Decision Table with Consolidation of Subsumable Decision Rules

What is the right delivery method for an order?							
rush order?	*any fragile item?*	*any specialty item?*	*any high-priced item?*	*any item involving hazardous materials?*	*category of customer?*	*destination of order?*	***delivery method for order***
no	*no*	*no*	*no*	—	*silver*	—	**picked up by customer**
yes	*yes*	*no*	*no*	*yes*	*gold*	*local*	**shipped by normal service**
yes	—	—	—	*yes*	*platinum*	*remote*	**shipped by premium service**

The two subsumable decision rules are absorbed into one in the contracted decision table illustrated in Figure P5-16.

This decision table shows the consolidated decision rule (highlighted in light purple in the top row) with a *does not matter* (dash) for the *hazardous materials* consideration.

5.10 CONTRACTION/EXPANSION OF DECISION TABLES

Contraction of a decision table is based on detecting and consolidating subsumptions (usually by software).

Expansion of a decision table does the opposite – more specific decision rules are shown individually even if subsumable.

Figure P5-17 illustrates. Observations:

- The expanded version of the decision table on the far left shows every possible combination (intersection case) of three binary considerations n, m and q, along with the appropriate outcome for each combination.
- The decision table in the middle highlights the subsumable decision rules on which contraction/expansion of the decision table can be based.
- The contracted version of the decision table is presented on the far right. A *does not matter* (dash) now appears for each elemental case in the rows that result from consolidation of the subsumable decision rules.

Additional Notes

- The set of intersection cases for all three versions of the decision table in Figure P5-17 is considered unique (i.e., to satisfy the single-hit principle).
- Contraction of a decision table can either be partial (i.e., only to more specific rules that have previously appeared), or complete (i.e., to all more specific rules possible, even if ones that have never appeared before).
- The contracted version of the decision table at the far right might be useful in spotting general rules. Does the bottom row in that decision table represent some deeper business intent? If so, an **outcome restriction** to give it *protection* might be specified. Refer to Part 5.11 for discussion.

Figure P5-17. Example of Contraction/Expansion of Decision Tables

expanded version

m	n	q	outcome
yes	yes	yes	a1
		no	a2
	no	yes	a2
		no	a2
no	yes	yes	a1
		no	a1
	no	yes	a1
		no	a1

expanded version showing subsumable decision rules

m	n	q	outcome
yes	yes	yes	a1
		no	a2
	no	yes	a2
		no	a2
no	yes	yes	a1
		no	a1
	no	yes	a1
		no	a1

contracted version

m	n	q	outcome
yes	yes	yes	a1
		no	a2
	no	—	a2
no	—	—	a1

5.11 USING OUTCOME RESTRICTIONS TO ENFORCE GENERAL RULES IN EXPANDED DECISION TABLES

An **outcome restriction** can be used to answer the special **pattern question**:

> *Does a* **general rule** *still apply even if it disappears in expanding a* **decision table**?

Refer to Part 5.10 for discussion of contraction/expansion of decision tables. Such manipulations typically involve the special symbol *does not matter* (dash). Refer to Appendix 7 for discussion of *does not matter* (dash) as used in **TableSpeak**.

In expansion of a decision table, general rules are no longer visible. Each is replaced by a set of more specific rule(s). In effect, each general rule is either partially or completely *un*subsumed.

The more specific rules can be individually modified, perhaps producing outcomes inconsistent with the original (but now invisible) general rule. Business intent can be lost in this fashion.

The outcome restriction *protected* can be specified for any general rule in a decision table. The restriction ensures the original business intent embodied in the general rule is retained.

Consequently, the general rule:

- Remains in effect even if it disappears in expanding a **decision table**.
- Is applied to all the more specific rule(s) produced from it.
- Dictates the outcomes all the more specific rule(s) produced from it. (The outcomes must all be the same as for the general rule.)

The pattern question above can thus be reworded:

> *Can a more specific rule produced through expansion of a decision table be modified (updated) directly such that its outcome is different than for the more general rule?*

To illustrate, let's return to the decision table presented in Figure P5-16, which was a contracted version of the decision table in Figure P5-15.

Consolidation of the two subsumable decision rules in the latter Figure resulted in the general

rule in the former Figure. Suppose *protection* is specified for that rule. Figure P5-18 illustrates.

This contracted decision table could be expanded (*un*subsumed) into the decision table shown in Figure P5-19.

- The two more specific rules have reappeared (in the top and bottom rows).
- The general rule is now hidden.

Even so, the *protection* specified for the general rule still applies. *Neither* of the two new more specific rules, both grayed-out, can be modified until such time that *protection* is discontinued, if ever. That original business intent is retained.

Figure P5-18. Decision Table with a Protected General Decision Rule

What is the right delivery method for an order?							
rush order?	*any fragile item?*	*any specialty item?*	*any high-priced item?*	*any item involving hazardous materials?*	*category of customer?*	*destination of order?*	***delivery method for order***
no	*no*	*no*	*no*	—	*silver*	—	**picked up by customer *(protected)***
yes	*yes*	*no*	*no*	*yes*	*gold*	*local*	**shipped by normal service**
yes	—	—	—	*yes*	*platinum*	*remote*	**shipped by premium service**

Figure P5-19. Expanded Decision Table with Hidden Protected General Rule

What is the right delivery method for an order?							
rush order?	*any fragile item?*	*any specialty item?*	*any high-priced item?*	*any item involving hazardous materials?*	*category of customer?*	*destination of order?*	***delivery method for order***
no	*no*	*no*	*no*	*no*	*silver*	—	**picked up by customer**
yes	*yes*	*no*	*no*	*yes*	*gold*	*local*	**shipped by normal service**
yes	—	—	—	*yes*	*platinum*	*remote*	**shipped by premium service**
no	*no*	*no*	*no*	*yes*	*silver*	—	**picked up by customer**

APPENDICES

APPENDIX 1: WHAT TABLESPEAK IS

TableSpeak is a set of conventions for business-friendly representation of **decision tables** and their meaning (semantics) in **declarative** fashion.

Central to TableSpeak are:

- Highly pragmatic guidelines and thresholds for selecting the best decision-table format. TableSpeak avoids simplistic and counterproductive one-size-fits-all approaches.
- The principle of **single-sourcing**, essential for achieving true business agility.
- **Restrictions** – **business rules** for ensuring the integrity (correctness) of decision-table content.

TableSpeak optimizes for readability by non-IT professionals and business people:

- The question the decision table answers is always emphasized.
- Scope (applicability) is declared explicitly.
- Unnecessary complications to decision-table structure (such as **exceptions**) are externalized.
- Meaning is comprehensively expressed.
- Business vocabulary is carefully used.

Decision tables based on TableSpeak are free of hidden assumptions and implicit interpretation semantics.

TableSpeak is fundamentally based on words (semantics) and explicit **rules**. It is suitable for representing **decision logic** anytime you need to know why you get the results you get *in words*. Refer to Appendix 2 in the companion Primer on **DecisionSpeak** for additional discussion of this guiding principle.[19]

[19] http://www.brsolutions.com/IPSpeakPrimers

APPENDIX 2: OTHER KINDS OF TABLES POTENTIALLY USEFUL IN ANALYZING OR COMMUNICATING DECISION LOGIC

Several kinds of tables other than **decision tables** are potentially useful in analyzing or communicating **decision logic**.

A2.1 LOOK-UP TABLES

In TableSpeak, a **look-up table** is a table whose content is populated by some mathematical formula.

The look-up table is provided as matter of convenience, so people can 'look up' the pre-computed result of a computation rather than compute it themselves on-the-spot. Tax tables, for example, are often look-up tables.

Table A-1 summarizes key differences between decision tables and look-up tables.

The formula for a look-up table is preferably expressed as a computation-type **business rule**.

> *Hint:* The formula for a look-up table often can be found in a footnote, at the end of the table, or in the accompanying instructions.
>
> Frequently the formula is given in the context of what to do if the **case** of interest falls outside the range(s) of the **consideration(s)** the table explicitly covers.

A2.2 SCENARIO TABLES

In TableSpeak, a **scenario table** (also called a **case table**) is a table whose content specifies **cases** relevant to an **operational business decision**, but not the appropriate **outcomes** for them.

A scenario table can be used to:

- Brainstorm outcomes.
- Explore what format of **decision table** works best for some **decision logic**.
- Discover **scope items**, **defaults**, **exceptions**, and **restrictions**.

A scenario table for the decision table in Figure P1-6 is presented in Figure A-1. For convenience the relevant **(intersection) cases** have been numbered and highlighted in purple.

Table A-1 Key Differences Between Decision Tables and Look-Up Tables

Differentiation Criteria	Decision Table	Look-Up Table
What is the primary purpose?	guide decisions	convenience
Can outcomes be produced by a formula?	no	yes
Should contents be directly updatable?	yes	no
What should be managed?	the decision table	the formula

Figure A-1. Scenario Table based on the *What coat should be worn?* Decision Table in the One-Rule-Per-Row Style of Figure P1-6.

Is it cold?	**Is it rainy?**	***Case***
yes	yes	*1*
	no	*2*
no	yes	*3*
	no	*4*

A scenario table for the decision table in Figure P1-1 is presented in Figure A-2. The relevant (intersection) cases have again been numbered for convenience and highlighted in purple.

Figure A-2. Scenario Table for the *What coat should be worn?* Decision Table in the Intersection Style of Figure P1-1.

		Is it rainy?	
		yes	***no***
Is it cold?	**yes**	1	*2*
	no	3	*4*

APPENDIX 3: ILLUSTRATION AND DISCUSSION OF ONE-RULE-PER-COLUMN DECISION TABLES

The **decision logic** for the *What is the right delivery method for an order?* **decision table** presented in Figure P2-7 in **one-rule-per-row** style is given in Figure A-3 as a **one-rule-per-column** decision table.

As preferred under **TableSpeak**, the **outcomes** for each of the **intersection cases** have been placed at the top of each column rather than the bottom.

delivery method for order, is shown at the top of each column (instead of to the right of each row).

- Each complete column (instead of each complete row) holds one **decision rule**. These rules could (again) be verbalized individually as in Appendix 4.

Figure A-3. One-Rule-Per-Column Decision Table for the Operational Business Decision *What is the right delivery method for an order?*

What is the right delivery method for an order?	*delivery method for order*		
	picked up by customer	shipped by normal service	shipped by premium service
rush order?	*no*	*yes*	*yes*
any fragile item?	*no*	*yes*	—
any specialty item?	*no*	*no*	—
any high-priced item?	*no*	*no*	—
any item involving hazardous materials?	*no*	*yes*	*yes*
category of customer?	*silver*	*gold*	*platinum*
destination of order?	—	*local*	*remote*

Observations

- Seven few-case considerations are listed from top to bottom in the left-most column (instead of along the top row).
- Three **intersection cases** are shown in each of three columns to the right (instead of in three rows).
- The appropriate **outcome** for each of the three intersection cases, the particular

A3.1 CAUTIONS USING THE ONE-RULE-PER-COLUMN STYLE

The one-rule-per-column style is not without certain dangers.

For example at first glance, Figure A-3 might seem complete because all three potential outcomes for *delivery method for order* are listed atop the three right columns.

But this decision table is far from complete! Refer to Appendix 5 for discussion.

Completeness in decision tables in *any* style is based on exhaustively addressing all elemental cases and all intersection cases, *not* simply addressing all potential outcomes at least once.

Quite often different decision rules, ones that cannot be subsumed, produce the very same outcome.

So anticipate that a one-rule-per-column decision table will show the same outcome more than once across the top (or bottom).

Should columns (decision rules) with the same outcome be shown in the decision table side-by-side?

That approach is generally not recommended except perhaps during initial capture. Lexicographical order is preferred overall. The TableSpeak prescription is:

> *Do nothing in the format of a decision table that will detract from achieving completeness with respect to elemental and intersection cases.*

APPENDIX 4: VERBALIZATION OF DECISION RULES

The three **decision rules** in the **one-rule-per-row decision table** in Figure A-4 can be individually verbalized as follows.[20]

Note: Figure A-4 originally appeared as Figure A2-7. Its **one-rule-per-column** counterpart appeared as Figure A-3.

Row 1 (Column 2 in Figure A-3):

The delivery method for an order must be 'picked up by customer' if all the following are true for the order:

- *It is not rush.*
- *It does not include any fragile item.*
- *It does not include any specialty item.*
- *It does not include any high-priced item.*
- *It does not include any item involving hazardous materials.*
- *The category of the customer that placed the order is 'silver'.*

Row 2 (Column 3 in Figure A-3):

The delivery method for an order must be 'shipped by normal service' if all the following are true for the order:

- *It is rush.*
- *It includes a fragile item.*
- *It does not include any specialty item.*
- *It does not include any high-priced item.*
- *It includes an item involving hazardous materials.*
- *The category of the customer that placed the order is 'gold'.*
- *The destination of the order is 'local'.*

Figure A-4. Decision Table in Figure P2-7

What is the right delivery method for an order?							
rush order?	*any fragile item?*	*any specialty item?*	*any high-priced item?*	*any item involving hazardous materials?*	*category of customer?*	*destination of order?*	***delivery method for order***
no	*no*	*no*	*no*	*no*	*silver*	—	**picked up by customer**
yes	*yes*	*no*	*no*	*yes*	*gold*	*local*	**shipped by normal service**
yes	—	—	—	*yes*	*platinum*	*remote*	**shipped by premium service**

[20] These verbalizations use the 'the following' clause in RuleSpeak. Refer to the *RuleSpeak Tabulation Primer:* http://www.brsolutions.com/IPSpeakPrimers

Row 3 (Column 4 in Figure A-3):

The delivery method for an order must be 'shipped by premium service' if all the following are true for the order:

- *It is rush.*
- *It includes an item involving hazardous materials.*
- *The category of the customer that placed the order is 'platinum'.*
- *The destination of the order is 'remote'.*

APPENDIX 5: SAMPLE ASSESSMENT OF THE COMPLETENESS OF A DECISION TABLE

The **true completeness** of any **decision table** cannot be determined without inspecting the underlying **structured business vocabulary** (**concept model**).

Suppose the business vocabulary for the decision table in Figure P2-7 indicates *semi-fragile* to be an additional **elemental case** alongside *fragile*. If so, the decision table is incomplete.

A5.1 APPARENT COMPLETENESS

The **apparent completeness** of a decision table – completeness without reference to underlying structured business vocabulary (concept model) – can be calculated mathematically (preferably by software).

For example, the apparent completeness of the decision table in Figure P2-7 can be calculated as follows (in three steps).

Step 1. The total number of *possible* **intersection cases** is calculated as follows.

- Six **considerations** apparently have two elemental cases each (*yes* and *no* for all except *destination of order,* which shows *local* and *remote*).
- The seventh, *category of customer*, apparently has three elemental cases (*silver*, *gold*, and *platinum*).

Therefore the total number of *possible* distinct **intersection cases** is calculated as: $2^6 \times 3 = 192$.

This total of course assumes that the business vocabulary confirms the actual number of possible elemental cases for *destination of order* and *category of customer* to be two and three, respectively.

Step 2. The total number of intersection cases *actually* covered in the current version of the decision table can be determined as follows.

One row, the second from the bottom, covers exactly one intersection case – each cell in that row has something in it other than a *does not matter* (dash).

The other two rows are a bit more complicated because both have one or more cells with *does not matter* (dashes). These rows represent **general rules**.

- The first (row just below the considerations) includes one such cell, so that row provides the **outcome** for *two* intersection cases – one if the cell had *local* and one if it had *remote*.
- The bottommost row includes three such cells, so that row provides the outcome for 2^3 or *8* intersection cases.

Altogether, the decision table therefore covers 11 distinct intersection cases (2 + 1 + 8 = 11).

Step 3. The apparent completeness of the decision table is calculated as follows.

The number of intersection cases the decision table does *not* cover is: 192 – 11 = 181. So 181 possible distinct intersection cases are not

addressed by either some **specific rule** or general rule.

Clearly this decision table is not very complete(!).

A5.2 DECISION RULE CAPTURE TABLE

The table in Figure P2-7 is so far from complete some practitioners would not call it a "decision table" at all.

Instead, it might be described as a decision rule *capture table* (sometimes *decision grid chart*) – simply a tabular list of certain intersection cases and associated outcomes.

A possible use could be to initially capture and specify some decision rules – perhaps to some tool.

APPENDIX 6: ILLUSTRATION OF THE NUMBER OF ROWS IN A ONE-RULE-PER-ROW-STYLE DECISION TABLE

In a **row-or-column-style decision table** every **decision rule** is represented by a row (or by a column).

How many rows (or columns) would be required for the *What should be charged for shipping an order?* decision table in Figure P2-3?

That **decision logic** includes three **many-case considerations**:

- There are approximately 43,000 *zip codes* in the United States.[22]
- Let's say the *weight* **consideration** involves 100 elemental cases.
- There are twelve *months*.

Multiplying 43,000 ×100 × 12 produces 51,600,000 rows (or columns)!

Notes

1. An **intersection-style** decision table for this same decision logic could also need that many **decision cells**. Such a decision table could be massive in its own way. An intersection-style decision table, however, would not have any redundancy of **elemental cases**.
2. Change in elemental cases is always a concern. For example, do *zip codes* ever change? In fact the total number of *zip codes* can fluctuate by a few thousand *annually*.[23] The ability to manage such change effectively is crucial.

For the sake of argument, let's say you find ways to consolidate the decision logic.

For example, for every decision rule you need to keep, let's say there are 999 you can eliminate because they produce the same **outcome**.

That's a reduction by three orders of magnitude to 51,600 rows or columns in the row-or-column-style. *You are going to need (good!) automation to represent and manage that many.*

[22] http://www.carrierroutes.com/ZIPCodes.html, available March, 2013

[23] ibid

APPENDIX 7: SPECIAL KEYWORDS AND SYMBOLS IN TABLESPEAK

TableSpeak uses a minimal set of special keywords and symbols whose meaning is explained below.

A7.1 ABOUT THE MEANING OF THE SPECIAL OUTCOME 'NONE'

The special keyword *none* appears in one of the **decision cells** of the **decision table** in Figure P1-1 (and other Figures).

None is obviously not a coat, but is a legitimate **outcome** nonetheless. A subtle implication is that the set of **potential outcomes** is broader (by one instance) than the actual class of the outcomes, *coat*.

None means no coat need be worn. It does *not* mean "the question cannot be decided" or "not known".

In other words, 'none' *does* provide a definitive outcome; it *does* answer the question the decision table addresses. No breach of scope occurs when *none* appears.

A7.2 ABOUT THE MEANING OF THE SPECIAL SYMBOL '—'

A dash in a cell of a decision table where an **elemental case** would otherwise appear means *does not matter*. The *does not matter* symbol (dash) is used widely in decision tables and is not unique to **TableSpeak**.

Used correctly a *does not matter* (dash) is a means by which the number of **intersection cases** in a decision table can be kept as low as possible.

In the **one-rule-per-row style** that means the fewest number of rows. Refer to Part 5.10 for discussion of contraction/expansion of decision tables.

A *does not matter* (dash) in a cell of a decision table indicates that the **decision logic** accepts *anything* there:

- *Any* elemental case.
- *Nothing at all*, which effectively means *unknown*.

If unknown is not deemed desirable, a **consideration restriction** should be specified. Refer to Part 5.8 for discussion.

In effect, a *does not matter* (dash) represents an implicit OR between all elemental cases of a given **consideration** (plus unknown). For example if the consideration is Boolean, the meaning of a dash is understood as *yes OR no OR* unknown.

Creating an intersection case with a *does not matter* (dash) for every consideration in a decision table is never meaningful or desirable.

A7.3 ABOUT THE MEANING OF THE SPECIAL TABLESPEAK OUTCOME 'N/A'

The TableSpeak symbol *n/a (not applicable)* indicates no outcome for a case is appropriate because the case has been specifically precluded by a restriction.

The symbol *n/a* can be viewed as definitive. In effect an *n/a* says, "This case is recognized as being a **case in scope**, but the correct answer to the question being asked is that there *is* no answer to the question in this case."

Cases indicated as *n/a* instead having a normal outcome are not **exceptions** of any kind – they do not use different considerations in their determination.

APPENDIX 8: STYLES OF DECISION TABLES IN PERSPECTIVE

The classic style of **decision table** in software engineering is called *limited entry decision table*. It was standardized in 1982.[24]

A related style is called *extended entry decision table*. Both styles are discussed and illustrated individually below.

A8.1 LIMITED ENTRY DECISION TABLES

The essential feature of the limited entry style is that **elemental cases** and **outcomes** – both called *entries* – are strictly limited to binary choices. (The reason presumably was partly to keep memory requirements to a minimum.)

The decision table from Figure P1-1 is illustrated in classic limited entry style in Figure A-5.[25] The symbols in the table mean:

C – condition

A – action

R – rule (condition-action)

X – 'do this'

Figure A-5. Figure P1-1 in Limited Entry Style

	Coat Table	R1	R2	R3
C1	raining	Y	Y	N
C2	cold	Y	N	Y
A1	wear lined raincoat	X		
A2	wear unlined raincoat		X	
A3	wear wool overcoat			X

Observations

- In Figure A-5 both elemental cases (*Y* and *N*) and outcomes (*X* and empty) are binary (and single characters) – hence the name *limited entry decision table*.
- The style is not at all business-friendly.

  Interpreting the decision table correctly requires distinguishing its four basic quadrants, and then, keeping all the special symbols in mind, essentially reading clockwise starting from the upper left quadrant.
- Since the binary restriction does not permit numbers (other than 0 or 1), limited entry decision tables have traditionally been closely associated with logic and formal reasoning, rather than broader business decision-making.
- The actions in the table are **procedural** and permit both real-world actions and program control actions (including the traditional programming *return* or *go back*). Generally a top-to-bottom 'execution' order is assumed.

[24] Codasyl, *A Modern Appraisal of Decision Tables, Report of the Decision Table Task Group*, ACM, New York, 1982

[25] Hurley, Richard B., *Decision Tables in Software Engineering*, Van Nostrand Reinhold, New York, NY, 1983, p. 8.

- 'Wear no coat' (*none*) is not present in the decision table because no action is required in that **case**.

A limited entry decision table can be purely **declarative**. As illustrated by Figure A-6 at least two revisions are appropriate:

- The *A* for *action* changed to *O* for the preferred TableSpeak term *outcome*.
- The imperative verb "wear" dropped from each outcome.

Figure A-6. Limited Entry Style Used Declaratively

	Coat Table	R1	R2	R3	R4
C1	raining?	Y	Y	N	N
C2	cold?	Y	N	Y	N
O1	lined raincoat	X			
O2	unlined raincoat		X		
O3	wool overcoat			X	
O4	none				X

Most experts deem the limited entry style simply outdated. It's also known to be subject to certain kinds of undesirable dependencies.

A8.2 EXTENDED ENTRY DECISION TABLES

The advantages of relaxing the restriction to binary 'entries' has long been recognized.

Extended entry decision tables permit elemental cases or outcomes (or both) to be non-binary.

An example of an extended entry decision table with non-binary elemental cases is given in Figure A-7, which is based on Figure P4-2.

Observations

- The consideration *temperature level?* is now not binary. It has three elemental cases – *cold, moderate* and *warm*.
- Single-character 'entries' *Y* and *N* for *precipitation?* have been replaced by the actual elemental cases – *rainy* and *not rainy.*

Note the large number of **decision cells** in Figure A-7. There are 36 (in light blue), most empty. And this is a relatively small example.

An obvious problem for this style of decision table is the large number of decision cells it can produce.

If there are *p* number of **intersection cases** and *q* number of **potential outcomes**, there can be as many as $p \times q$ number of decision cells. (There are $6 \times 6 = 36$ in Figure A-7.)

The actual number of decision cells might be considerably smaller of course if the decision table is consolidated through contraction. Refer

Figure A-7. Extended Entry Decision Table with Non-Binary Elemental Cases

Coat Table

temperature level?	cold		moderate		warm	
precipitation?	rainy	not rainy	rainy	not rainy	rainy	not rainy
lined raincoat	X					
wool overcoat		X				
unlined raincoat			X			
sweater				X		
umbrella					X	
none						X

to Part 5.9 for a discussion and illustration of contraction (and expansion).

In any case, the number of potential outcomes obviously needs to be kept relatively low in this style. Some practitioners would permit 10 to 20 potential outcomes; a more reasonable threshold is probably 4 or 5.

Although the style used in Figure A-7 might be useful for certain problems, especially logic-intensive ones, **TableSpeak** generally does not recommend it for business use.

A8.3 THE ROOTS OF ALL ROW-OR-COLUMN-STYLE DECISION TABLES

All limited entry and extended entry decision tables are special cases of the **one-rule-per-column** style.

This fact becomes obvious when *both* elemental cases and outcomes are 'extended'. Figure A-8, a revision of Figure A-7, illustrates.

In Figure A-8 the six outcomes have been 'extended' – i.e., indicated as outcomes directly in the decision table (instead of indirectly, using *X*'s as in Figure A-7). The original 36 decision cells have now been reduced to six.

Remember that the one-rule-per-column and **one-rule-per-row** styles are really just variations of the same basic approach, the **row-or-column-style**.

Does it become clear now that the roots of this style lie in software?

Figure A-8. The Decision Table in Figure A-7 in One-Rule-Per-Column Style

Coat Table

temperature level? / precipitation?

cold		moderate		warm	
rainy	not rainy	rainy	not rainy	rainy	not rainy
lined raincoat	wool overcoat	unlined raincoat	sweater	umbrella	none

REFERENCES

The 'Speak' Series

Decision Analysis – A Primer: How to Use DecisionSpeak™ and Question Charts (Q-Charts™)
by Ronald G. Ross
available May, 2013 (free) on http://www.brsolutions.com/IPSpeakPrimers

Decision Tables – A Primer: How to Use TableSpeak™
by Ronald G. Ross
available May, 2013 (free) on http://www.brsolutions.com/IPSpeakPrimers

Tabulation of Lists in RuleSpeak®: A Primer – Using "The Following" Clause
by Ronald G. Ross
available May, 2013 (free) on http://www.brsolutions.com/IPSpeakPrimers

On good decision tables and normalization …

Knowledge Factoring Using Normalization Theory
by Vanthienen J and Snoeck M
International Symposium on the Management of Industrial and Corporate Knowledge (ISMICK) 1993, pp. 97 – 106

Rules as Data: Decision Tables and Relational Databases
by Jan Vanthienen
Business Rules Journal, Vol. 11, No. 1 (Jan. 2010) URL: http://www.BRCommunity.com/a2010/b516.html

On what is wrong with bad decision tables …

50 Ways to Represent Your Rule Sets
by Jan Vanthienen
Business Rules Journal, Vol. 7, No. 1 (Jan, 2006) URL: http://www.BRCommunity.com/a2006/b266.html

Best practices …

Simplifying and Optimizing Tabular Decision Models
by Jan Vanthienen
Business Rules Journal, Vol. 12, No. 10 (Oct. 2011) URL:
http://www.BRCommunity.com/a2011/b618.html

Modeling Decision Table Structures
by Jan Vanthienen
Business Rules Journal, Vol. 11, No. 11 (Nov. 2010) URL:
http://www.BRCommunity.com/a2010/b565.html

For a general overview …

The History of Modeling Decisions using Tables (Part 3)
by Jan Vanthienen
Business Rules Journal, Vol. 13, No. 5 (May 2012) URL: http://www.BRCommunity.com/a2012/b652.html

The History of Modeling Decisions using Tables (Part 2)
by Jan Vanthienen
Business Rules Journal, Vol. 13, No. 3 (Mar. 2012) URL: http://www.BRCommunity.com/a2012/b641.html

The History of Modeling Decisions using Tables (Part 1)
by Jan Vanthienen
Business Rules Journal, Vol. 13, No. 2 (Feb. 2012) URL: http://www.BRCommunity.com/a2012/b637.html

Decision Tables: An Overview of the Existing Literature
by Moreno A, Verhelle M and Vanthienen J

GIACA, Information Technologies in Accounting, Fifth International Conference on Artificial Intelligence and Emerging Technologies in Accounting, Finance and Tax (Huelva, Spain 2000)
URL: http://www.econ.kuleuven.ac.be/prologa/download/overview82-2000.pdf

About RuleSpeak (English, Spanish, Dutch, German) …

www.RuleSpeak.com

On business rules and concept models …

Business Rule Concepts: Getting to the Point of Knowledge (4th ed.)
by Ronald G. Ross, 2013
http://www.brsolutions.com/b_concepts.php

Business Rules Manifesto: The Principles of Rule Independence
by Business Rules Group, 2003
URL: www.BusinessRulesGroup.org/brmanifesto.htm

Semantics of Business Vocabulary and Business Rules (SBVR)
Object Management Group (OMG)
For an introduction and explanation, refer the *SBVR Insider* section on www.BRCommunity.com/SBVR

Early standardization …

Codasyl, *A Modern Appraisal of Decision Tables, Report of the Decision Table Task Group*, ACM, New York, 1982

Decision tables in software engineering …

Hurley, Richard B., *Decision Tables in Software Engineering*, Van Nostrand Reinhold, New York, NY, 1983, 164pp.

GLOSSARY

Term	Definition	Example(s)[26]	Note(s)	Dictionary Basis[27]
action	that which someone or something does	A. Put clothes on B. Determine sales tax		*3: the process of doing : exertion of energy*
anomaly	irregularity or abnormality		See **conflict**, **omission**, **redundancy**, and **subsumption**.	*3: something irregular or abnormal* *[anomalous]3b: exhibiting or containing incongruous or often contradictory elements*
apparent completeness	**completeness** based only on the representation of some **decision logic** (e.g., as a **decision table**), not also on the **structured business vocabulary** (**concept model**) that underlies it		**Apparent completeness** can be assessed mathematically.	
behavioral rule	a **business rule** indicating an obligation concerning conduct, action, practice, or procedure; a **business rule** whose purpose is to shape (govern) day-to-day business activity and prevent undesirable situations (states) that could occur at any of various points in time	A service representative must be assigned to a customer that has placed an order.	1. From the OMG standard *Semantics of Business Vocabulary and Business Rules* (SBVR). 2. **Operational business decisions** are always based on **decision rules**, not **behavioral rules**. 3. Contrast to **decision rule**. 4. Refer to ***Business Rule Concepts: Getting to the Point of Knowledge*** (4th ed, 2013), by Ronald G. Ross.	
business rule	a **rule** that is under business jurisdiction		From the OMG standard *Semantics of Business Vocabulary and Business Rules* (SBVR).	
case	a particular situation	A1. It's rainy. A2. It's cold and rainy and it's a workday. B. A purchase is made in Harris	1. A **case** is simply some matter arising in day-to-day business operations. **Cases** are of interest in **decision analysis** when an **operational business**	*1b: a set of circumstances constituting a problem: a matter for consideration or*

[26] Examples designated "A" and "B" pertain to aspects of the same problem in multiple entries.

[27] All definitions are from *Merriam-Webster Unabridged Dictionary*.

Term	Definition	Example(s)[26]	Note(s)	Dictionary Basis[27]
		County during 2013.	**decision** needs to be made. 2. Think of a **case** as a snapshot of circumstances that momentarily at least don't 'flow' – i.e., as something **declarative** rather than **procedural**. (For that reason the term **case** is preferred over the term **scenario**.) 3. **Case** (particular situation) is to **consideration** (factor in making an **operational business decision**) as *instance* is to *class.*	*decision: as (1): a circumstance or situation*
case in scope	any **case** that satisfies the **considerations** used to establish scope for an **operational business decision**		1. **Decision logic** can handle only **cases** within scope. Other **cases** must be handed off (to some expert, manager, process, or other **decision logic**). 2. **Decision logic** should be able to give **outcomes** for all **cases** provably within scope. 3. **Cases in scope** include both **standard cases** and **exceptional cases** (if any).	
case table	a table whose content specifies **cases** relevant to an **operational business decision**, but not the appropriate **outcomes** for them		A **case table** (sometimes called a **scenario table**) can be used to: Brainstorm **outcomes**. Explore what format of **decision table** works best for some **decision logic**. Discover **scope items**, **exceptions**, **general rules**, **defaults**, and **restrictions**.	
caveat (relevance restriction)	a **relevance restriction** that warns that if any **outcome** is provided for some **case(s)** the **outcome** cannot be considered necessarily valid	For the **operational business decision** *What premium should an applicant be charged?* a **caveat** might be *Securing coverage at the given price subject to eligibility.*	Contrast to **pre-emption.**	
certainty of outcome (in decision tables)	a quality goal for **decision tables** in which **cases** always produce some **outcome(s)** that is/are definitively correct		**Certainty of outcome** in **decision tables** can be endangered by: • Overlapping (non-**disjoint**) or redundant **elemental cases.** • Redundant **intersection cases**. • **Multiple outcomes for decision rules**.	

Term	Definition	Example(s)[26]	Note(s)	Dictionary Basis[27]
			• **Omissions** (missing **decision rules**). • **Conflicts.** • Violations of **restrictions.**	
completeness	the state of **decision logic** in which there are no missing **rules** (**omissions**) for any **case in scope**		1. See also: **apparent completeness, true completeness.** 2. See also **omission**. 3. Not all **intersection cases** need necessarily appear in a **decision table** for it to be **complete**; for example if *does not matter* (dash) appears in the **decision table**. 4. (Some) **default(s)** can be specified for a **decision table** to guarantee **completeness**. 5. Special allowance for **omissions** is made in the **sparse style**.	*[complete - adjective] 1a: possessing all necessary parts, items, components, or elements : not lacking anything necessary : ENTIRE, PERFECT*
complexity threshold	in **decision logic**, the number of **considerations** at which the representation by a **decision table** generally becomes impractical		1. The **complexity threshold** is exceeded if either of the following is true: • The total number of **many-case considerations** exceeds 3. • The total number of all **considerations** exceeds 7. 2. The **complexity threshold** is one of three fundamental style-related thresholds in **TableSpeak**. See also **row-or-column threshold** and **multi-table threshold**.	
concept model	the semantic blueprint of a **structured business vocabulary**		Refer to ***Business Rule Concepts: Getting to the Point of Knowledge*** (4th ed, 2013), by Ronald G. Ross.	
ConceptSpeak™	the Business Rule Solutions, LLC (BRS) set of conventions, guidelines and techniques for representing **operational business decisions** in business-friendly fashion, diagramming **decision structures**, and coordinating them with **structured business vocabulary** (**concept models**)		1. For more about **ConceptSpeak** refer to the chapters on concept models in ***Business Rule Concepts: Getting to the Point of Knowledge*** (4th ed, 2013), by Ronald G. Ross. 2. **ConceptSpeak** is part of **IPSpeak**.	

Term	Definition	Example(s)[26]	Note(s)	Dictionary Basis[27]
conflict (decision logic)	an anomaly in **decision logic** where two **decision rules** share the same **considerations** and cover exactly the same **cases**, but have mutually-exclusive **outcomes**	For the operational business decision *What is the right delivery method for an order?* the potential outcome *picked up by customer* is mutually exclusive with the potential outcome *shipped by normal service*. (If an order is picked up it can't be shipped, and vice versa.) If the same case leads to both outcomes, a conflict arises.	The prevention, detection and elimination of **conflicts** are important activities in creating high-quality **decision logic**.	
consideration	a factor in making an **operational business decision**; something that can be resolved to two or more **cases**	A1. Whether it is cold. A2. Whether it is a work day. B1. Which county is a purchase made in. B2. What year is a purchase made in.	1. A **consideration** can be posed as an individual question to be answered; however, **DecisionSpeak** does not require this. 2. Also known as *condition*. **Consideration** is preferred in **DecisionSpeak** because it is more business-friendly and intuitive.	*[consideration] 3a: something that is considered as a ground of opinion or action* *[consider] 1: to reflect on : think about with a degree of care or caution*
consideration dependency	one **operational business decision** being dependent on the **outcome** of another **operational business decision** such that the **outcome** of the latter **decision** provides or supports one of the **considerations** for the dependent **decision**	You can't decide what to wear unless you decide whether it's cold.	1. Note on the example: For certain **cases** relevant to a given **operational business decision** (e.g., *What should be worn?*), the appropriate **outcome** (e.g., what to wear) depends on some **consideration(s)** that can be resolved only by evaluating the **decision logic** for another **operational business decision** (e.g., *Is it cold?*). So deciding whether it is cold (based on appropriate **considerations**) is a prerequisite for determining what to wear. 2. Contrast with **relevance dependency** and **outcome dependency**.	
consideration restriction	a **restriction** that precludes certain ways in which **considerations** can be combined	For the operational business decision *Where should breakfast be eaten?* a **consideration restriction** for the *desire to cook?* consideration might	1. Contrast with **relevance dependency** and **outcome dependency**. 2. A **consideration dependency** can be used to protect against unknowns – i.e., the situation where an elemental case is missing from an intersection	

Term	Definition	Example(s)[26]	Note(s)	Dictionary Basis[27]
		be *n/a if traveling*.	case when *does not matter* (dash) is used.	
crosstab (representation) style (of decision tables)	*see* **intersection-style**			
decision	a determination requiring **know-how** or expertise; the resolving of a question by identifying some correct or optimal choice		Contrast to **operational business decision**.	*[decision] 1b: a determination arrived at after consideration* *[determination] 2: the resolving of a question by argument or reasoning* *[decide]; to dispel doubt on* *a: to arrive at a choice or solution concerning which ends uncertainty or contention* *c: to infer or conclude from available indications and evidence*
decision analysis	identifying and analyzing some key question arising in day-to-day business activity and capturing the **decision logic** used to answer the question		1. **Decision analysis** focuses on **operational business decisions** in day-to-day business activity whose answers need to be determined, inferred or concluded. 2. The deliverable of **decision analysis** is **decision logic** in the form of **decision tables**, **business rule** statements, and **Q-Charts** that are: • deployment-ready • anomaly-free • business friendly	
decision box	a means to organize specifications pertaining to one or more **decision tables**		1. A **decision box** can substitute for a **wrapper rule statement**. 2. A **decision box** contains the question the **decision logic** answers, a list of the **considerations**, and possibly other specifications such as **scope item(s), default(s)**, **exception(s)**, and **restriction(s)**.	

Term	Definition	Example(s)[26]	Note(s)	Dictionary Basis[27]
decision cell	a cell in a **decision table** where an **outcome** appears			
decision logic	the set of all **decision rules** for **cases in scope**	For every county in Texas, the applicable sales tax for every year, as well as any **exceptions**.	1. So that determinations made in **operational business decisions** can be consistent, traceable, manageable, and repeatable, **decision logic** should be captured and represented in the form of: • **decision table(s)** • **business rule** statements(s) • some combination thereof. 2. **Decision logic** includes **decision rules** for both **standard cases** and **exceptional cases**.	
decision rule	a **business rule** that guides the making of an **operational business decision**; specifically, a **business rule** that links a **case** to some appropriate **outcome**	A. *A wool suit must be worn on a cold workday when it isn't raining.* B. *The applicable sales tax for a purchase must be 8.25% in Harris County during 2011.*	1. A **decision rule** provides a specific answer to a selective question. 2. A **decision table** represents a collection of **decision rules**. 3. Although usually not necessary, any **decision rule** represented in a **decision table** can also be written as a textual statement. 4. Contrast to **behavioral rule**.	
decision-smart business process	a business process in which **decision tasks** are recognized explicitly and their **decision logic** is externalized		A **decision-smart business process** is one whose **decision logic** is agile.	
DecisionSpeak™	the Business Rule Solutions, LLC (BRS) set of conventions, guidelines and techniques for defining terms, designing a **concept model**, and developing a **structured business vocabulary**		1. Refer to ***Decision Analysis – A Primer: How to Use DecisionSpeak™ and Question Charts (Q-Charts™)***, available (free) on http://www.brsolutions.com/IPSpeakPrimers. 2. Central to **DecisionSpeak** is expressing the questions that **operational business decisions** address. 3. **Q-COEs** and **Q-Charts** are part of **DecisionSpeak**. 4. **DecisionSpeak** optimizes for readability by non-IT professionals and business people. • Sequential dependencies are removed in favor of logical	

Term	Definition	Example(s)[26]	Note(s)	Dictionary Basis[27]
			dependencies. • **Structured business vocabulary** is always carefully used. No compromises are made, however, in the accuracy and completeness of the **decision rule(s)** with respect to the subject matter. There is never any hidden or assumed interpretation semantics. 5. **DecisionSpeak** is fundamentally based on words (semantics) and explicit **rules**. It is suitable for **decision analysis** anytime you need to know why you get the results you get *in words*. 6. **DecisionSpeak** is part of **IPSpeak**. 7. The natural follow-on to **DecisionSpeak** is **TableSpeak**, with which **DecisionSpeak** is closely aligned.	
decision structure	how one or more **operational business decisions** are formally organized		1. A **decision structure** should indicate **operational business decisions** that are dependent – i.e., **consideration**-dependent, relevance-dependent, and/or **outcome**-dependent. 2. A **Q-Chart** visually organizes a **decision structure**.	
decision table	a structured means of visualizing **decision rules** in rows and columns		1. A **decision table** identifies the appropriate **outcome** for each **case** it covers. 2. In a true **decision table**, **outcomes** cannot be predicted by a formula. If the **outcomes** could, there would be no need for the **decision table**. 3. Contrast with **look-up table**.	

Term	Definition	Example(s)[26]	Note(s)	Dictionary Basis[27]
decision task	a task centered on making an **operational business decision** – that is on deciding something rather than on doing something	A. Determine clothes to wear B. Determine sales tax on purchases	1. As recommended for all tasks, a **decision task** should be named using a verb + noun (or noun phrase). 2. A **decision task** always involves a determination of some kind – hence, **DecisionSpeak** recommends using the convention "*Determine …*" to name a **decision task**. 3. A **decision task** often appears in a business process model (but does not have to).	
declarative	constituting an expression or representation that can be either true or false		1. There is no implicit meaning (semantics) in **declarative** expressions or representations. 2. Contrast to **procedural**. 3. In graduate school in the early 1970s (the days of punch cards) I learned a highly pragmatic test for determining whether specifications are **declarative**: • Take each statement of the specifications and type it on an individual punch card. • Assemble the deck. • Test it to make sure it works. • Throw the whole deck up in the air. • Pick up all the cards in random order. • Re-test it. If the logic still works, the statements are **declarative**. If not, they are **procedural**. The point is that in **declarative** specifications no logic is lost 'between the lines' – i.e., none is intrinsic to the sequence of presentation. In other words, there is no hidden meaning (semantics).	*2: constituting a statement that can be either true or false*
default (decision rule)	a **decision rule** or set of **decision rules** that supplies the appropriate **outcome(s)** for **omissions** in a **decision table**		1. **Defaults** are an important means by which **decision tables** can be simplified and kept as small as possible. 2. A **default** supplies some appropriate **outcome(s)** if a **decision table** does not address some **case(s)** explicitly. 3. (Some) **default(s)** can be specified for a **decision table**	

Term	Definition	Example(s)[26]	Note(s)	Dictionary Basis[27]
			to guarantee **completeness**. 4. A **default** and an **exception** are not the same thing. A **default** uses the 'normal' **considerations** for a **decision table**; an **exceptional case** never does. 5. **Defaults** can be as simple as a single quantity or instance, or as complex as a separate **decision table**. 6. **Defaults** should be used with caution. A **default** might leave important **cases** unexamined that should have distinct **outcomes** – especially as **decision logic** evolves over time. Once (re)deployed, such un(re-)examined **decision logic** can produce unpleasant surprises. 7. Specifying some **default(s)** for common or usual **outcome(s)** can permit a **decision table** to be represented in **sparse style** without any gaps in the **decision logic**.	
disjoint (cases)	not overlapping	For the **consideration** *Is it cold?* … 1. The **cases:** • *Yes, it's cold.* • *No, it's not cold.* are *disjoint.* 2. The **cases:** • *Below 11°C* • *Above 9°C* are not *disjoint.*	A **decision table** with overlapping **cases** can produce uncertain **outcomes**. In other words, overlapping **cases** endangers **certainty of outcome**.	*2 : having no members in common*
elemental case	a **case** produced from a single **consideration**	The elemental **cases** produced by the single **consideration** *Is it cold?* are: • *Yes, it's cold.* • *No, it's not cold.*	Contrast with **intersection case**.	
embedding (of a consideration in a decision table)	in a **decision table**, **elemental cases** for one **consideration** being repeated within every **elemental case** of another **consideration**		A **row-or-column-style decision table** always features **embedding**.	
exception	*see* **exceptional case**			

Term	Definition	Example(s)[26]	Note(s)	Dictionary Basis[27]
exceptional case	a **case in scope** that does not use the **considerations** of a **standard case**; i.e., a **case in scope** that is based on some **consideration(s)** that is/are not among the **considerations** for a **standard case**	A. Halloween B. Food C. Auto insurance: • An applicant is convicted of a felony involving a motor vehicle. • An applicant for auto insurance is not 18 years of age or older. • The applicant is the boss's daughter.	1. Special **decision logic** must be developed for each **exceptional case**. 2. The **decision logic** for an **exceptional case** might be as simple as a single **decision rule** (e.g., *The boss's daughter must be accepted for auto insurance.*), or significantly more complex. 3. A **default** and an **exception** are not the same thing. A **default** uses the 'normal' **considerations** for a **decision table**; an **exceptional case** never does.	*[exception] 2: one that is excepted or taken out from others* *[except - transitive verb] 1: to take or leave out (something) from a number or a whole : exclude or omit*
exhaustive	covering all possible **cases in scope**	For the **consideration** *Is it cold?* … 1. The **cases:** • *Yes, it's cold.* • *No, it's not cold.* are **exhaustive.** 2. The **cases:** • *colder than 11°C* • *11°C ≤ temperature ≤ 27°C* are not **exhaustive.**	1. A **decision table** with missing **elemental cases in scope** is likely to be incomplete. So is a **decision table** with missing **intersection cases**. 2. A decision table can exhibit no **redundancy** with respect to **intersection cases** without being **exhaustive** – and vice versa.	
few-case consideration	a **consideration** that involves only two or three **elemental cases**	• *true? (yes, no)* • *fragile? (fragile, not fragile)* • *state of matter (solid, liquid, gas)*	1. How many **elemental cases** a **consideration** produces is an important factor in selecting the best **decision table** format. 2. Contrast to **many-case consideration**. 3. Also called a **simple consideration**.	
general rule	a **decision rule** that covers more than one **case**	1. All shipments to Hawaii must cost the same, regardless of *season, weight, packaging, etc.* 2. All shipments to Alaska in the winter cost the same, regardless of *weight, packaging, etc.*	1. A **general rule** covers many **cases** in a single specification, treating all such cases uniformly. 2. A **general rule** often produces an **outcome** based on a proper subset of relevant **considerations**. 3. Contrast with **specific rule**.	
gray-out (of cells in a decision table)	the **TableSpeak** convention indicating that no revision (updating) is permitted for a cell in a **decision table**		1. Revision of **grayed-out** cells is permitted only after the **restriction** is removed (if ever). Until then the **grayed-out** cells are effectively locked. 2. Any alternative means of	

Term	Definition	Example(s)[26]	Note(s)	Dictionary Basis[27]
	because a **restriction** applies		visually indicating which cell(s) in a **decision table** cannot be revised is acceptable.	
independent subdecision	one of a collection of two or more **operational business decisions** on which another **operational business decision** is dependent via **consideration dependencies**	Determining the overall eligibility of a candidate for auto insurance depends on the **independent subdecisions**: • *Eligible according to proof of insurance?* • *Eligible according the driving history?* • *Eligible according to risk score?* • etc.	1. Each subdecision in the collection may be evaluated (a) separately, and (b) in parallel or in any sequence. 2. Each subdecision has a distinct **outcome** and a different set of **considerations** (usually non-overlapping) from its peers in the collection.	
inductive reasoning	reasoning from a part to a whole, from particulars to generals, or from the individual to the universal			*[induction] 2b(1)*
intersection case	a **case** representing a combination of one **elemental case** from each of two or more **considerations**	The **intersection cases** produced by combining the **considerations** • *Is it cold?* • *Is it rainy?* are: • *It is cold and rainy.* • *It is cold and not rainy.* • *It is not cold and not rainy.* • *It is not cold but rainy.*	Contrast with **elemental case**.	
intersection-style (format of decision table)	using both columns and rows of a **decision table** to represent two or more **considerations** and their **elemental cases** with appropriate **outcomes** in intersection cells		1. Contrast with **row-or-column-style (format of decision table)**. 2. Sometimes called **crosstab (representation) style.**	
IPSpeak™	the Business Rule Solutions, LLC (BRS) methodology for capturing, expressing, analyzing, and managing operational-level intellectual		1. Creating and managing operational-level intellectual property (IP) is what provides competitive advantage for the organization and makes it smart.	

Term	Definition	Example(s)[26]	Note(s)	Dictionary Basis[27]
	property (IP) of the business, specifically **business rules**, **decision logic**, **structured business vocabulary** (**concept models**), and strategy		2. **IPSpeak** includes **RuleSpeak**, **DecisionSpeak**, **TableSpeak**, **ConceptSpeak**™, and StrategySpeak™. 3. For more about **business rules**, **concept models**, and **ConceptSpeak** refer to ***Business Rule Concepts: Getting to the Point of Knowledge*** (4th ed, 2013) by Ronald G. Ross. 4. For more about capturing **business rules** and about strategy refer to ***Building Business Solutions: Business Analysis with Business Rules***, by Ronald G. Ross with Gladys S.W. Lam, 2011.	
know-how	accumulated practical skill or expertness; especially technical knowledge, ability, skill, or expertness of this sort			*... accumulated practical skill or expertness ...* *especially: technical knowledge, ability, skill, or expertness of this sort*
logical dependency (in business rules)	an expression or representation of **business rules** using the name of something computed or derived by another business rule	Rule 1. *A rush order must not be suspended.* Rule 2. *An order is always considered rush if the delivery time due is less than 24 hours from the time placed.* Rule 1 is logically dependent on rule 2.	1. In **business rules**, a **logical dependency** is always based on some element(s) of a **structured business vocabulary** (**concept model**), not sequence. 2. Use of **logical dependencies** is an acceptable and highly desirable means to **single-source** different bits of **business-rule** logic.	
look-up table (in decision logic)	a table whose content is populated by some mathematical formula	*Tax tables are often look-up tables.*	1. A **look-up table** is not a true **decision table**, in which **outcomes** cannot be predicted by a formula. (If the **outcomes** could, there would be no need for the **decision table**.) 2. The production of a **look-up table** is purely a matter of convenience, so that people can "look up" the result of a computation rather than compute it themselves. 2. The mathematical formula for a **look-up table** is preferably expressed as a computation-type **business rule**. 3. For the purposes of **business**	

Term	Definition	Example(s)[26]	Note(s)	Dictionary Basis[27]
			rule management, manage the formula, not the **look-up table** itself. 4. The **outcomes** in a **look-up table** should never be directly updatable. Rather, they should always be populated only via application of the formula.	
many-case consideration	a **consideration** that produces four or more individual **cases**	• *weight* • *date of birth* • *zip code* • *month*	1. How many **elemental cases** a **consideration** produces is an important factor in selecting the best **decision table** format. 2. Contrast to **few-case consideration**.	
multiple outcomes (for a decision rule)	the same **case** having two or more **outcomes** in one or more **decision rules**		**Multiple outcomes** are of special concern if any of the following are true. The **outcomes**: • Belong to different classes, suggesting they answer different questions. • Are not explicitly or'ed or and'ed, leaving doubt about which apply.	
multi-table (representation for decision tables)	in the **intersection style**, using multiple tables to represent **decision logic**		1. The easiest way to think of a three-dimensional array is as multiple two-dimensional arrays (tables). 2. The number of two-dimensional tables required to represent three **considerations** is equal to the number of **elemental cases** for the third **consideration**. 3. The number of two-dimensional tables required to represent *four* **considerations** is equal to the number of **elemental cases** for the third **consideration** multiplied by the number of **elemental cases** for the fourth **consideration** (and so on). 4. See also **multi-table threshold.**	
multi-table threshold	in the **intersection style**, the number of **many-case considerations** (three) at which multiple tables are apparently required to represent **decision logic**		1. *Beyond* three **many-case considerations**, the practitioner is well-advised to reconsider the **operational business decision** by analyzing the following questions: • Can some portion(s) of the **outcomes** be computed by	

Term	Definition	Example(s)[26]	Note(s)	Dictionary Basis[27]
			some formula(e)? • Can the **operational business decision** be decomposed into subdecisions? • Is it really that important to manage the **decision logic** *in words*? • Is the selectivity in **outcomes** really worthwhile to the business? 2. The **multi-table threshold** is one of three fundamental style-related thresholds in **TableSpeak**. See also **complexity threshold** and **row-or-column threshold**.	
omission (in decision logic)	an **anomaly** in **decision logic** where some **case in scope** is not addressed by any **rule**		1. An **omission** is literally a missing **rule**, which in turn indicates **decision logic** is not **complete**. See also **completeness**. 2. Special allowance for **omissions** is made in the **sparse style**.	
one-rule-per-cell (format of decision table)	see **intersection-style (format of decision table)**			
one-rule-per-column (format of decision table)	the **row-or-column-style** that uses only columns to represent entire **decision rules**			
one-rule-per-row (format of decision table)	the **row-or-column-style** that uses only rows to represent entire **decision rules**			
operational business decision	a determination requiring operational business **know-how** or expertise; the resolving of an **operational business question** by identifying some correct or optimal choice	A. *What coat should be worn?* B. *What sales tax needs to be paid on a purchase?*	1. Contrast to **decision**. 2. In **DecisionSpeak**, an **operational business decision** is always identified and analyzed from the business point of view, not IT. 3. An **operational business decision** can always be posed as a question. **DecisionSpeak** prescribes using this question as the name for the **operational business decision**. 4. An **operational business decision** should always satisfy the DOORS criteria (deterministic, operational, objective, repetitive, single-point of determination).	

Term	Definition	Example(s)[26]	Note(s)	Dictionary Basis[27]
			5. An **operational business decision** involves determining the appropriate answer (**outcome**) for a **case** from among all **potential outcomes**.	
operational task	a task centered on doing something rather than on deciding something	• Take customer order. • Notify applicant of acceptance.	An **operational task** should involve nothing significant to decide, no significant choice between alternatives to make, just some **actions**.	
outcome	a **potential outcome** that is deemed appropriate for some **case**	A. *wool suit* B. *8.25%*	1. An **outcome** is the result, conclusion or answer given by a **decision rule** to a selective question being asked. 2. **Decisions** addressed by **decision analysis** generally feature exactly one **outcome** for each **case**. 3. **Outcomes** are often single terms or values, but may also be mathematical formulas, natural-language expressions or sentences, etc. 4. Also known as *conclusion*. **Outcome** is preferred in **DecisionSpeak** because it is more business-friendly and intuitive.	*[conclusion] 1a: a reasoned judgment or an expression of one : INFERENCE*
outcome dependency	one **operational business decision** being dependent on the **outcome** of another **operational business decision** such that the **outcome** of the latter **decision** dictates some **outcome(s)** of the dependent **decision**	The **operational business decision** *What should be charged for shipping an order?,* is **outcome**-dependent on the **operational business decision** *What set charges are there for shipping an order?*.	1. In an **outcome dependency**, kinds of **outcome** for the respective **operational business decisions** must align. 2. Contrast with **consideration dependency** and **relevance dependency**.	
outcome restriction	a **restriction** that limits certain **cases** to a particular **outcome**, or to some subset of all **potential outcomes**	For the operational business decision *What should be charged for shipping an order?,* an **outcome restriction** might be *$250 for zip code 99950 & weight < 5 kg*.	1. 1. Contrast with **relevance dependency** and **consideration dependency**. 2. An **outcome restriction** reduces or eliminates choice of **outcome**.	
pattern question	a thinking tool that assists in developing **business rules**, especially **behavioral rules**, from business design artifacts (e.g.,		1. Each **pattern question** focuses on a particular topical concern and some particular construct (pattern) found frequently in artifacts of a given kind. The questions are	

Term	Definition	Example(s)[26]	Note(s)	Dictionary Basis[27]
	business process models, **concept models**, etc.)		designed to assist in asking the right kinds of questions in the right ways to capture **business rules**. 2. Refer to ***Building Business Solutions: Business Analysis with Business Rules***, by Ronald G. Ross with Gladys S.W. Lam, 2011.	
point of determination	a point in operational business activity where one or more **business rules** are evaluated or applied in a particular matter		All the **decision rules** for an **operational business decision** (e.g., *What should be charged for shipping an order?*) would likely be applied only at a single **point of determination** (e.g., when an order is taken). A **behavioral rule** in contrast generally should be applied at multiple **points of determination**. Consider the **behavioral rule** *A student with a failing grade must not be an active member of a sports team.* This **business rule** does not apply only at a single **point of determination** (e.g., when a student joins a team). Instead, the **business rule** is meant to be enforced continuously – for example, if a student who is already active on some sports team should let his or her grades fall.	
potential outcome	some result, conclusion, or answer that might be deemed appropriate for some **case** addressed by some **decision**	A. *wool suit, dress, pants,* etc. B. *Applicable sales tax* could be *8.25%, 7%, 9.5%*, etc.		
practicable	an element of guidance (e.g., **business rule**) that is ready to deploy into business operations such that it can satisfy the following test: [it is] sufficiently detailed and precise that a person who knows [about it] can apply it effectively and consistently in relevant circumstances to	The statement *Safety is our first concern. …* is not **practicable**. The statement (**business rule**) *A hardhat must be worn on the head in a construction site. …* is **practicable**.	1. From the OMG standard *Semantics of Business Vocabulary and Business Rules* (SBVR). 2. The results for a **practicable** statement should be the same no matter whether deployed to staff or ultimately to machines.	*1: possible to practice or perform : capable of being put into practice, done, or accomplished* *2a: capable of being used : USABLE*

Term	Definition	Example(s)[26]	Note(s)	Dictionary Basis[27]
	know what behavior is acceptable or not, or how something is understood			
preempted case	a **case** for which no **outcome** is meaningful		A **relevance restriction** can produce **preempted cases**.	
preemption (relevance restriction)	a **relevance restriction** that precludes providing any **outcome(s)** for some **case(s)**	For the **operational business decision** *What should be charged for shipping an order?* **preemptions** might be: *1. zip codes 99928 thru 99929* *2. tubes to zip code 04402*	Contrast with **caveat**.	
preemption list	a list set up specifically to identify **preempted cases** in a **decision table**		A **preemption list** is a special form of **relevance restriction** where a long list of **cases** is required (typically four or more).	
preemption table	a **decision table** set up specifically to identify **preempted cases** in another **decision table**		1. **A preemption table** is a special form of **relevance restriction** typically set up when **preemptions** are highly specific or selective and would be cumbersome to specify individually. 2. The set of **considerations** for a **preemption table** must be either the same as for the other **decision table**, or a subset. 3. The **sparse style** is sometimes desirable for representing **preemption tables**.	
procedural	an expression or representation meant to be included in a series of other expressions or representations to specify a procedure		1. There is meaning (semantics) implicit in the sequence that the expressions or representations of a procedure are indicated. 2. Contrast to **declarative**.	*[procedure]1b(3): a series of steps followed in a regular orderly definite way*
Q-Chart™	a visualization of some **decision structure**		1. The "Q" stands for 'question' – that is, the question(s) the **decision structure** addresses. 2. A **Q-Chart** organizes **Q-COEs**. 3. See also **DecisionSpeak**.	
Q-COE™	a graphic representation of an		1. **Q-COEs** can be used on their own for brainstorming, or	

Term	Definition	Example(s)[26]	Note(s)	Dictionary Basis[27]
	operational business decision indicating what question ('Q') is being asked, and possibly one or more of the following: • **considerations** ('C') • **outcomes** ('O') • **exceptions** ('E)		included in **Q-Charts**. 2. See also **DecisionSpeak**.	
question chart	*see* **Q-Chart**			
redundancy (of decision rules)	an **anomaly** in a **decision table** where the same **decision rule** appears more than once		1. If uniqueness of **elemental cases** for each **consideration** is ensured, an **intersection-style decision table** is physically resistant to **redundancy of decision rules**. 2. **Redundancy** of **decision rules** in **row-or-column-style decision tables** can generally be avoided by software that ensures uniqueness of all **intersection cases**. 3. A decision table can exhibit no **redundancy** with respect to **intersection cases** without being **exhaustive** – and vice versa. 4. See also **single-hit**, which provides a higher level of protection against **redundancies**.	
relevance dependency	one **operational business decision** being dependent on the **outcome** of another **operational business decision** such that the **outcome** of this other **decision** may eliminate the need for any **outcome** from the dependent **decision**	1. If you decide not to have breakfast, you don't need to decide where to get it. 2. If a company decides not to ship anything to Puerto Rico, then it does not need to determine the cost of shipping there.	1. For certain **cases**, the less dependent **operational business decision** (e.g., *Can an order be shipped to a location?*) can preempt a dependent **operational business decision** (e.g., *How much does it cost to ship to a location?*). In those **cases**, the dependent **decision** becomes meaningless. 2. Contrast with **consideration dependency** and **outcome dependency**.	
relevance restriction	a **restriction** that either: • precludes providing any **outcome(s)** for some **case(s)** **(preemption)** • warns that if any **outcome** is provided for some **case(s)** the	For the **operational business decision** *What should be charged for shipping an order?* **relevance restrictions** might be the **preemptions**: *1. zip codes 99928*	1. Contrast with **consideration dependency** and **outcome dependency**. 2. A **relevance restriction** can produce **preempted cases**.	

Term	Definition	Example(s)[26]	Note(s)	Dictionary Basis[27]
	outcome cannot be considered necessarily valid – **(caveat)**	*thru 99929* *2. tubes to zip code 04402*		
restriction (for decision table)	a **business rule** that directly governs the integrity (correctness) of a **decision table**		1. A **restriction** acts as a limitation on how a **decision table** may be populated with **decision rules**. It pertains to managing the decision table over time. 2. **Restrictions** often permit **single-sourcing** within **decision tables** and thus retention of business intent.	
row-or-column-style (format of decision table)	using only the physical columns or the physical rows of a **decision table** to represent all **considerations** and their **elemental cases** plus appropriate **outcomes** – i.e., entire **decision rules**		1. The **row-or-column-style** always unites **elemental cases** in a series (i.e., effectively concatenated) using either rows or columns. 2. See also **one-rule-per-*column*** and **one-rule-per-*row* (formats of decision table).** 3. The **row-or-column-style** might also be called the *single-axis style* since it always uses one or the other of the following – *never both* –to display **decision rules**: • The vertical (y) axis to designate rows for **decision rules**. • The horizontal (x) axis to designate columns for **decision rules.** 4. Contrast with **intersection-style (format of decision table)**.	
row-or-column threshold	in **decision tables**, the number of **considerations** (six) at which the **intersection style** generally becomes impractical and the **row-or-column-style** should be used to represent **decision logic** instead		The **row-or-column threshold** is one of three fundamental style-related thresholds in **TableSpeak**. See also **complexity threshold** and **multi-table threshold**.	
rule	a guide for conduct or action; a standard on which a **decision** or judgment may be based		From the OMG standard *Semantics of Business Vocabulary and Business Rules* (SBVR).	*1a: guide for conduct or action* *1f: one of a set of usually official regulations by which an activity (as a sport) is governed e.g., *the*

Term	Definition	Example(s)[26]	Note(s)	Dictionary Basis[27]
				infield fly rule *the rules of professional basketball* *['criteria'] 2a: standard on which a decision or judgment may be based*
rule independence	the externalization, unification, and management of **business rules** separately from processes		1. Refer to ***Business Rule Concepts: Getting to the Point of Knowledge*** (4th ed, 2013), by Ronald G. Ross. 2. Refer to the Business Rules Manifesto, http://www.businessrulesgroup.org/brmanifesto.htm	
RuleSpeak®	the Business Rule Solutions, LLC (BRS) set of guidelines and conventions for expressing **business rules** in concise, business-friendly fashion using structured natural language		1. Refer to www.BRSolutions.com for more about **RuleSpeak** (free). 2. **RuleSpeak** is part of **IPSpeak**.	
scenario	*see* **case**		**Case** is preferred over **scenario** because of the latter term's connection with events and thus activity. **Case**, in contrast, simply means *situation*, with no connotation of activity.	*a sequence of events especially when imagined; especially: an account or synopsis of a possible course of action or events*
scenario table	*see* **case table**		See the note for **scenario**.	
scope item (decision table)	a factor along with some explicit **cases** identified externally to a **decision table** to which the **decision table** is deemed applicable		1. A **decision table** should not be applied except as permitted by its **scope items** (if any). If it is, the result is **inductive reasoning** – generally not appropriate for **business rules**. 1. An implicit 'and' is always assumed for **scope items**. 2. **Scope items** are one technique used under **TableSpeak** to keep **decision tables** as simple as possible	
sequential dependency (of questions)	a question applying only if another question asked previously has a certain answer		Sequential dependencies arise naturally in logic trees, flow charts, and other procedural models.	
simple consideration	*see* **few-case consideration**			
single-hit	ensuring strict logical		1. The **single-hit principle**:	

Term	Definition	Example(s)[26]	Note(s)	Dictionary Basis[27]
(principle for decision tables)	uniqueness of **intersection cases** in **row-or-column-style decision tables**		• Treats a *does not matter* (dash) as a logical expression to be evaluated, not just as its own **elemental case**. • Examines the OR expressions resulting from each *does not matter* (dash) to test uniqueness of **intersection cases**. • Ensures *logical* uniqueness of **intersection cases**, not just physical uniqueness. • Achieves a higher degree of protection against **anomalies** (redundancies) than physical uniqueness alone. 2. Supporting the **single-hit principle** generally requires software. 3. See also **redundancy**.	
single-sourcing	specifying a **business rule** only once no matter how many places are affected (e.g., cells in a **decision table**) or the **rule** is deployed		1. **Single-sourcing** provides a single point of change and coordination for business practices. 2. **Single-sourcing** should be interpreted as applicable within a defined scope (usually not enterprise-level).	
smart business process	a **decision-smart business process** in which violation actions are specified for **behavioral rules** separately, rather than embedded in the process itself		A **smart business process** is an all-around agile process.	
sparse (style of decision table)	showing only **intersection cases** that are associated with some specific, relatively rare **outcome(s)**		1. In the **sparse style** of **decision table** the usual prescription against missing **intersection cases** / **omissions** are relaxed. Only **intersection cases** with some specific, relatively rare **outcome(s)** (e.g., an *X, no,* etc.) are represented so that those specific **cases** are highly visible. 2. So that there are no gaps in **decision logic**, some **default(s)** for the common or usual **outcome(s)** can be specified when the **sparse style** is used. 3. The **sparse style** is	

Term	Definition	Example(s)[26]	Note(s)	Dictionary Basis[27]
			sometimes desirable for representing **preemption tables**.	
specific rule	a **decision rule** that covers only one **case**	Determining the cost to ship to a specific *zip code*, say *77005*, requires using all considerations for the operational business decision – e.g., *season, weight, packaging,* etc.	1. A **specific rule** always produces an **outcome** based on the full set of relevant **considerations**. 2. Contrast with **general rule**.	
standard case	a **case in scope** that is regular or common, and cannot be excluded from normal treatment or rejected out-of-hand	A. It's a *cold, rainy* workday. B. A purchase is made in *Harris County* during *2011*.	**Standard cases** generally make up the bulk of **cases in scope**.	*[standard - adjective] 3a: regularly and widely available : readily supplied : not unusual or special*
structured business vocabulary	the set of terms and their definitions, and all wordings, that organize operational business **know-how**		Refer to ***Business Rule Concepts: Getting to the Point of Knowledge*** (4th ed, 2013), by Ronald G. Ross.	
subsumption (among individual business rules)	an **anomaly** among individual **business rules** that all require or disallow the same state, except that one **business rule** requires or disallows the state for a superset of **cases** than the other **business rule**	First **business rule**: *A rush order must have a destination.* Second **business rule**: *An order must have a destination.* The first **business rule** is subsumed by the second **business rule**.	The prevention, detection and elimination of **subsumptions** are important activities in creating high-quality **business rules.**	*[subsume] 1 : to view, list, or rate as component in an overall or more comprehensive classification, summation, or synthesis : encompass as a part, example, or phase : classify as part of a larger schema or judge as a specific instance governed by a general principle*
subsumption (in decision tables)	a circumstance in a **decision table** where two **decision rules** share the same **considerations** and have the same **outcome**, but one **decision rule** covers a subset of the **cases** covered by the other **decision rule**		An instance of **subsumption in decision tables** can result from either of the following: 1. Simplification of the **decision logic** for visualization, usually by a software tool. A **decision table** can be contracted (made smaller) by subsuming all possible **decision rules**. 2. Accidental specification, usually by humans. Any such **subsumption** is an **anomaly**.	
TableSpeak™	the Business Rule Solutions, LLC (BRS) set of conventions, guidelines and		1. Refer to ***Decision Tables – A Primer: How to Use TableSpeak™***, available (free) on	

Term	Definition	Example(s)[26]	Note(s)	Dictionary Basis[27]
	techniques for representing **decision tables** in the most business-friendly fashion		http://www.brsolutions.com/IPSpeakPrimers. 2. **TableSpeak** shows how to express the meaning (semantics) of **decision tables** and to identify applicable restrictions. 3. **TableSpeak** is part of **IPSpeak**. 4. **TableSpeak** is closely aligned with **DecisionSpeak** and is **DecisionSpeak**'s natural follow-on.	
true completeness	**completeness** based on both the representation of **decision logic** (e.g., as a **decision table**) and the **structured business vocabulary** (**concept model**) that underlies it		**True completeness** depends on both: (a) alignment of the meaning of a question with that of its **considerations**, and (b) whether any **elemental cases** otherwise known for the **considerations** are absent. Both can be assessed only by reference to the underlying **structured business vocabulary** (**concept model**).	
wrapper rule (statement)	the semantics (meaning) of a **decision table** expressed in the form of a rule that indicates how the **decision table** should be interpreted – i.e., how to 'read' it		A **wrapper rule** requires a meaningful name to be given to the targeted **decision table** for the **rule** to reference the **decision table**.	

ABOUT THE AUTHOR: RONALD G. ROSS

Ronald G. Ross is Principal and Co-Founder of Business Rule Solutions, LLC, where he actively develops and applies the IPSpeak™ methodology including RuleSpeak®, DecisionSpeak™ and TableSpeak™.

Ron is recognized internationally as the "father of business rules." He is the author of ten professional books including the groundbreaking first book on business rules *The Business Rule Book* in 1994. His newest is *Building Business Solutions: Business Analysis with Business Rules* with Gladys S.W. Lam (2011, An IIBA® Sponsored Handbook).

Ron serves as Executive Editor of BRCommunity.com and its flagship publication, *Business Rules Journal*. He is a sought-after speaker at conferences world-wide. More than 50,000 people have heard him speak; many more have attended his seminars and read his books.

Ron has served as Chair of the annual International Business Rules & Decisions Forum conference since 1997. He was a charter member of the Business Rules Group (BRG) in the 1980s, and an editor of its *Business Rules Manifesto*. He is active in OMG standards development, with core involvement in SBVR.

Ron holds a BA from Rice University and an MS in information science from Illinois Institute of Technology. For more information about Mr. Ross, visit www.RonRoss.info which hosts his blog.

Follow Ron's tweets on Ronald_G_Ross.

ABOUT BUSINESS RULE SOLUTIONS

Business Rule Solutions, LLC

Business Rule Solutions, LLC is the recognized world leader in the advancement of business rules and decision management. Co-Founders Ronald G. Ross and Gladys S.W. Lam are internationally acclaimed as the foremost experts and practitioners of related techniques and methodology.

Since its inception in 1996, BRS has helped pilot the worldwide growth of business rules. BRS offers IPSpeak™, its groundbreaking methodology for business rules, decision logic, and business vocabulary (concept models), including the popular RuleSpeak®, DecisionSpeak™, TableSpeak™ and ConceptSpeak™. BRS services include consulting, on-line interactive training, in-house workshops, publications, and presentations. For more information about BRS, visit www.BRSolutions.com.