

# Resource Systems in Games: An Analytical Approach

Tróndur Justinussen, Peter Hald Rasmussen, Alessandro Canossa and Julian Togelius

**Abstract—** We describe an approach to using standard data mining algorithms to help analyse recurring themes in game design across several games, and to help suggest novel game design ideas. This is illustrated with the analysis of 119 different resource systems across 20 games. Clustering is used to validate the assignment of resources into archetypes; frequent pattern mining is used to find commonly co-occurring resource attributes; and decision tree induction is used to visualize the relations between resource archetypes. We discuss the relation between qualitative and quantitative analysis of game design and suggest that qualitative analysis is necessary but that quantitative methods can be of invaluable help.

## I. INTRODUCTION

Data mining in games is a trending and growing topic both within the academic games research community and within data analytics departments of commercial game development companies. Most current efforts are focused on understanding players of games. Many modern networked games produce vast amounts of log data about their players, and this data can be mined to understand the players in much the same way as customer analytics is used to understand customers. Examples of this kind of game analytics include the recent use of unsupervised learning to identify player types in Tomb Raider Underworld, and the use of supervised learning methods to predict player behaviour in the same game [1], [2]. Additionally, data mining techniques have been used to predict properties of the player beyond the game, for example the player's personality [3], [4]. Data mining techniques have also been used to groups of players in World of Warcraft [5], [6], [7].

However, data mining can be applied not only to the players, but also to the games themselves. In particular, this could be done for the purposes of assisting game designers and game design researchers. There are many thousands of published games, and the proven strengths of computational intelligence methods for finding patterns across multiple instances could help identify patterns in game design and suggest new avenues for creative work.

In this paper, we describe the application of several well-known tools from computational intelligence to help analyse a particular feature of games: Their resource systems.

Gathering, managing and spending resources is an integral part of game play. Yet, very little research has been done to formalize these important game elements. Björk & Holopainen[8] created a framework for patterns in games in general, and did not focus exclusively on resource systems.

When presented with resources such as health, mana, money and ammunition, players immediately have a notion of the dynamics of the resource based on past experience. This is because patterns in dynamics exist between resources, even across very different games. The term dynamics refers to the game-play that emerges from the player's interaction with the resource system. Similar to the term described in the MDA [9].

A framework has been created that is able to deconstruct individual resources into a number of attributes, that combined describe the resource pattern.

The resource systems have been encoded in an instance-attribute form, and clustering and frequent pattern mining have been used to identify archetypes and recurring themes. The results of the qualitative analysis is compared with the outcomes of the clustering process, and we discuss how the incongruences can help both illuminate the qualitative analysis of resource archetypes and suggest revisions to the attributes used in the encoding. Decision tree analysis has also been done in order to help visualize the relations between qualitatively discovered archetypes.

The entire process of creating and utilizing the framework could not be fully described in this paper. The process is described in detail in our thesis [10].

## II. INTRODUCING THE FRAMEWORK

A framework has been created through which resources systems in games can be analysed. Resource systems have been defined by the authors as part of the analysis process as: *“game elements controlled by the player that are gained, stored and spent to fund actions in order to achieve a goal or taken away as penalty. Their purpose is clear to the player, and they are valuable because of their scarcity.”* Part of this definition is inspired by previous work on game typology [11].

Creating such a framework to analyse resources in games has several prospects:

- To help designers create diverse and novel resource systems, and ultimately, create better games.
- Only through formalizing the fundamental aspects of resources can in-depth discussion of resource systems take place, which is the first step of creating an intrinsic language for formally discussing resources.
- It may lead to better predictions of the effects resource systems have on gameplay.

The framework is a systematic analysis tool that in a simple way describes resource dynamics by breaking the dynamics of the resource down into a combination of properties and attributes, which together form a full description of the resource.

The logic of the framework is based on classical Aristotelian categorization theory where objects are separated from other objects by defining properties that most accurately describe their features.

Each property represents areas of the dynamics of a resource. To each property belong a number of attributes that determine the value of the property. Simpler put, each property asks a question about a specific dynamic of a resource, and attributes represent the possible answers asked by the property. Each property can contain none or several attributes. Properties and attributes are designed to be inclusive enough to describe all resources, while still being able to differentiate between the dynamics of each different resource.

The framework includes 44 different attributes spread across 10 properties (see Table I). These have been created through extensive analysis of 119 resource systems found in 20 different games. The games were selected across a wide variety of genres, and based on the uniqueness of the resource systems in them. The resource systems of each game have been analysed, and properties and attributes have been created to adequately describe the dynamics of each individual resource.

As the framework is based on only 119 resources, it is not comprehensive enough to describe all conceivable resource systems, but the framework is expansive; more resources, properties and attributes can be added to the framework. The more there are added, the more comprehensive and descriptive the framework becomes. Also, existing properties and attributes can be modified as a consequence of adding more resources. The number of properties and attributes in the framework determines its descriptive power, but also its complexity; as the number of properties goes up, resources become better described, but at the cost of readability. As the framework is meant to be used as a design tool, the number of properties and attributes needs to strike the right balance between complexity and usability. The 44 attributes distributed in 10 properties appear to

adequately describe the 119 resources on which the framework is based.

The analysis of the different resources using the framework is not intended to describe all facets about that particular resource to such a degree that it can be recreated from scratch based purely on a unique combination of attributes. Such a framework would be too complex to be used as a design tool. Instead, the framework is meant to describe the resources in a sufficient enough degree for a designer to understand its function in the game in which the resource exists.

### III. READING THE FRAMEWORK

The framework is divided into three stages: Gain, store and spend. These stages represent the life cycle of resources. All resources are first gained by the player, then stored, and lastly spent in order to fund an action. (See Fig. 1)

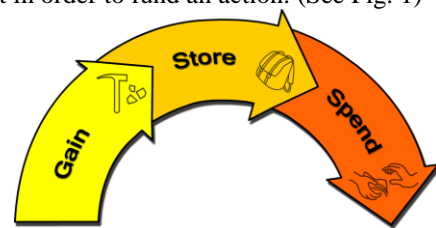


Fig. 1: The resource life cycle. Resources always undergo three stages; gain, store and spend.

Properties in the gain stage of the life cycle deal with how the resource is acquired by the player, and ask the questions: How is the resource gained from outside the game world? How is the resource gained from within the game world? Through which game element is the resource collected?

Properties in the store stage ask the questions: How much of the resource can be in the player's possession at a time? In which form is the resource is stored?

Properties in the spend stage ask the questions: What conversion options are there? What spending options are there? What is produced? What is the purpose? What is the long term consequence of spending the resource?

Through these narrowing questions, the vital information of the resource is condensed into manageable keywords that refer to its own intrinsic language.

TABLE I  
PROPERTIES FRAMEWORK

Gain			Store		Spend				
Non-Spatial	Generator	Focus Loci	Storage Cap	Entity	Conversion	Options	Produces	Purpose	Consequence
Regenerates	Enemy (Kill)	Unit	Limited	Virtual	Trade	Multi	Object	Enhance	Diminish
Non-Temporal	Being (Action)	Avatar	Unlimited	Object	Craft	Single	Effect	Inflict	Constant
RM	Pick-Up	God's Finger		Abstract	Exchange	Degenerates	Abstract		Invest
Reward	Recharger				Consumable	Lost	Unit		Vital
Starting	Lim. Vendor								
Ally	Vendor								
Exchange	Mine								
Consumable	Supplier								
Craft									
Trade									

This table includes all the possible properties and specifications that can be used to describe resources.

TABLE II  
SAMPLE RESOURCE ANALYSIS OF AMMUNITION IN CRYSIS 2

	Gain			Store		Spend					
Crysis 2	Non-Spatial	Generator	Focus Loci	Storage cap	Entity	Conversion	Options	Produces	Purpose	Consequence	Archetype
Ammunition		Enemy(Kill) + Pick-Up	Avatar	Limited	Virtual		Single	Effect	Inflict	Diminish	Ammo

This table describes how a resource would be described using the model. Each property (Generator, Focus Loci, Avatar etc.) contains some attributes that describe the behavior of the resource.

Table II shows an example of the framework being used to describe the ammunition resource in Crysis 2, a First Person Shooter game. The framework describes that the resource is gained from killing enemies and picked up from single-use containers in the game world through an avatar.

The player can only store a limited amount of ammunition at any one time, and the resource is presented as a virtual entity, meaning it is presented to the player as a number. Ammunition type resources have a single specific purpose. When spent, it inflicts a diminishing effect on an enemy, mostly in the form of direct damage.

Attribute and property names refer to an internal vocabulary, and as such work as keywords that refer to a well-established definition of each aspect of resource dynamics. This allows the framework to represent a lot of data using simple keywords.

Although the framework refer to its own intrinsic language, the vocabulary is in part based on obvious choices that intuitively describe the dynamics the attributes represent, but it is also, where possible, based on already established discourse. Mainly from *Patterns in Game Design*, Björk & Holopainen[5].

#### IV. APPLICATION

The framework reveals dynamics of existing resources. This facilitates the creation of novel resource systems that break traditional dynamics. There are several methods of using the framework as a design tool to modify existing or creating novel resources. Four examples are:

1) *Attribute Swapping Method*: This method applies the framework to an existing resource, and then add, remove or swap out attributes. Each attribute that is added or removed alters the dynamics of the resource.

2) *New Attributes Method*: The list of current attributes can be used as inspiration for creating new attributes not yet represented in the framework, that describe so far unseen dynamics.

3) *Randomization Method*: Combining a random combination of attributes can yield unpredictable and novel dynamics.

4) *Breaking the Framework Method*: The model is built on an intrinsic set of rules based on existing resource systems. Breaking these rules may reveal so far unexplored ways of perceiving resources.

As the framework is open-ended, these methods are only suggestions, and not an exhaustive list.

#### V. ARCHETYPES

Analysing a wide variety of games revealed easily recognizable recurring dynamics among the resource systems found in the games. This discovery meant that recurring resource dynamics across games could be mapped and categorized via attributes and properties, because resources that exhibit similar dynamics have a similar combination of attributes. These groups of similar resources were named archetypes.

Each of the 119 resource systems found in the analysis were categorised into 11 different archetypes, save for a few outliers. These categorisations of dynamics describe and distinguish between different game dynamics that emerge from each different resource system.

Each archetype is defined by a combination of attributes that a resource must have in order to belong to a certain archetype. Not all attributes need to be shared among resources within an archetype, only a key combination of attributes is needed.

The 119 resources were categorized into 11 archetypes, each with a satisfactory different combination of attributes and with a recognizable identity.

The dynamics of each archetype is described in the archetype framework through attributes that were found to always appear in the resources analysed (see Table III). For instance, 15 instances of Ammo resources were found in the 20 analysed games. These 15 resources share five attributes: Generator, limited, effect, inflict and diminish. These five attributes reveal a big chunk of the dynamics of Ammo. The shared attribute combination of Ammo resources are that they are always collected from resource generators from the game world, never collected by units, only a limited amount of Ammo can be stored at any one time, and have an inflicting, diminishing effect.

The undefined attributes in each archetype mean that those attributes of its members are different from one another, and are not needed to be shared to be a member of the archetype.

The validity of each archetype is evaluated based on the number of shared attributes by its members. A rule was made that at least half the properties must be shared by all members of the archetype in order for the archetype to exist. And the archetype must have at least three members in order for the archetype to be viable. By expanding the framework, the requirements for creating additional mutually exclusive archetypes is likely to increase.

In order to enhance the descriptive power of the archetype framework, it also weighs the importance of some attributes higher than others. These were named key attributes. These are marked with a “+” in the archetype framework (see table III). Key attributes indicate that the attribute is the most important identifier for the archetype in comparison to other

archetypes. The attributes marked with a “÷” explain that the absence of the attribute is an important identifier for the archetype. This form of categorization is entirely based on the qualitative expectations of the dynamics of the resources in each Archetype. For instance, the most important identifier for Mana is that it regenerates over time, and for Ammo that inflicts damage to an enemy.

The method for defining archetypes was to first identify resource dynamics that were recurring across several games. Archetypes such as Mana, Health and Currency were expected to exist and were therefore quickly identified. These were the resources with the most easily recognizable dynamics. The initial archetypes were grouped together based on a priori expectations of each resource’s dynamics. Attributes were then created to describe each individual archetype, and separate it from the remaining archetypes. For instance, the first attribute created was regenerates. Regenerates means that a resource regenerates on its own over time. All resources that regenerate over time were then grouped together into the Mana archetype. The name Mana was selected for the archetype due to convention. Mana exists in many games, and most players would expect Mana to be regenerating. Regenerates then became the key attribute for Mana.

Once the initial set of archetypes was defined and separated, the attributes created to describe them was applied to a small group of other resources. Most of these resources could not be properly identified through the current available attributes, so new attributes were created that describe the new resources, while still being applicable to the already existing archetypes. This was an iterative process, where a few new resources were added to the framework in each iteration.

At the end of the analysis process, 11 archetypes had been created. These can be described as follows:

1. *Health*: (19 resources) – Health is centered on the vital attribute. All resources that make the player or units under the player’s control die once exhausted are categorized into Health.
2. *Currency*: (18 resources) – Currency contains all resources that exist in recognizable buy/sell patterns, where things are being bought or sold for money.
3. *Ammo*: (15 resources) – Ammo is centered on the inflict attribute, as it is something that always inflicts a negative effect on opposing entity, such as ammunition.
4. *Mana*: (13 resources) – Mana is centered on the

regenerates attribute, and is typically being used to fuel actions or ‘spells’ in Role-Playing Games.

5. *Potion*: (13 resources) – Potions include items that induce an immediate enhancing effect on the player and are then removed from the game.
6. *Skill Point*: (12 resources) – Skill Point are points earned from progression that augment the player’s abilities.
7. *Rage*: (9 resources): Rage is gained from performing certain actions against enemy units or avatars in the game world, to fund more powerful actions.
8. *Material*: (7 resources): Materials are used to create objects in the game world or abstract items in the player’s inventory.
9. *Action*: (4 resources): Actions are a limited resource that can be spent on a fixed number of actions each turn or level.
10. *Physical*: (3 resources): Physical includes resources that are not stored but exist in the game world, and are spent to make new objects in the game world.
11. *Rune*: (3 resources): Runes are items that provide a long-term bonus that augment the player’s avatar’s abilities. Similar to skill Points, but Runes are items that have a single fixed effect and can be traded.

In addition to these archetypes, three resources were found that were too different from other resources to belong to any archetype. These were named outliers.

Expanding the framework to include more games will likely break some of the existing archetypes. But the framework is flexible enough to allow archetypes to be split up and form new archetypes by creating additional descriptive attributes and properties to describe new similarities between resources.

The existence of archetypes suggests that easily recognizable recurring dynamics of resources are to be found across games, and even among very different genres. This shows that resources across different games and genres share enough similarities to be analysed as a whole. Archetypes are useful when creating an intrinsic language for resource discussion, as archetypes, in a single word, represent large groups of resources and their dynamics. Moreover, archetypes suggest that a standardized tool can be used to analyse all conceivable resource systems. The data can even be structured in a way that allows it to be analysed using formal analysis such as clustering and decision trees to further explore resource systems.

TABLE III  
ARCHETYPE FRAMEWORK

	Gain			Store		Spend				
	Non-Spatial	Generator	Focus Loci	Storage cap	Entity	Conversion	Options	Produces	Purpose	Consequence
Action	+ Non-Temp. + Starting	÷ Generator		Limited	Virtual		Single		Enhance	
Ammo		Generator	÷ Unit	Limited				Effect	+ Inflict	Diminish
Currency				+ Unlimited	Virtual		Multi		Enhance	Constant
Health	Starting			Limited	Virtual	- None -	Lost			+ Vital
Mana	+ Regenerates			Limited	Virtual	÷ Trade		Effect		Diminish
Material		Generator	Avatar		+ Abstract	Trade	Multi	÷ Virtual	Enhance	Constant
Physical		Generator			÷ Virtual		Single	+ Object	Enhance	Constant
Potion		Generator	Avatar	Limited		Conversion	÷ Multi		Enhance	+ Diminish
Rage		+ Being (Action)	Avatar	Limited	Virtual	- None -		Effect		Diminish
Skill Point	+ Reward			Unlimited	Virtual	- None -	Multi	Effect	Enhance	Constant
Rune	Craft				Abstract	Craft	Single	Effect	Enhance	+ Constant

In this table, each resource archetype is defined using a combination of attributes. A resource system must contain each attribute in order to belong to any of the existing archetypes.

## VI. CLUSTERING

Clustering algorithms [12] were utilized on the resource property framework (see Table I) from the initial qualitative analysis. These could bypass the initial assumptions of archetypes that the qualitative analysis was based on, and produce less biased categories of resources. These could then be inspected and compared to the previously defined archetypes and function as an external validation of the qualitative analysis.

The systematized resource definitions from the qualitative analysis of the game resources first went through a pre-processing phase. The qualitative resource analysis was structured in a way that reflected its purpose as a design and analysis framework for game developers. This included the use of grouping attributes. For instance, the conversion property could contain multiple attributes that described the ways in which a resource could be spent by a player in order to generate other resources. This meant that the conversion property could contain many different combinations of attributes. Whereas the multi-value properties were meant to denote the collection of relevant attributes, data mining tools would read each property as a combination of all the attributes denoted, resulting in discrepancies in the automated analyses. Therefore, the multi-value properties of the original model were divided, making each of the 44 different attributes into a separate property column. The resulting data set was a table comprised of 44 columns containing binary values that each described the presence or absence of one particular attribute in the resource. This defined a possible solution space of 2 to the power of 44 ( $1.7592186 \times 10^{13}$ ) different combinations.

The resulting CSV file was imported into Weka 3.6.6 [12] for further analysis.

Clustering was done using the Weka SimpleKMeans clustering algorithm. This algorithm was chosen over other algorithms such as the Fuzzy C-Means, because it mirrored the classical categorization method used in the qualitative analysis.

Analyses of Dendrograms showed interesting groupings at low cluster counts, but also revealed that when the number of clusters exceeded 12, groupings with a population of only 1 or 2 data points started appearing. These outliers were not deemed interesting in this context, so it was decided to concentrate on the lower cluster counts.

Several cluster analyses were done using a varying amount of clusters in order to do comparison analysis between the groupings, to observe the division of clusters and watch the emergence of new resource clusters as the cluster count increased. More importantly, the generated cluster centroid attributes were compared to the archetypes definitions created in the qualitative resource categorization in order to find where the results overlapped and where they differed significantly.

The qualitatively created archetypes used a design-oriented structure when defining the requirements for inclusion of a resource. This included using:

- 1) *Inclusive properties* (resource must contain attribute X)
- 2) *Exclusive properties* (resource must not contain attribute Y)
- 3) *Selective properties* (resource must contain one or more from a list of attributes)
- 4) *Ignored properties* (values of the property are ignored when evaluating a resource)
- 5) *Key properties* that were considered important for the archetype.

By contrast, KMeans clustering algorithms use standard distance functions to define the clusters, and output centroids described with a specific attribute combination. This meant that no direct comparison could be done when comparing the found clusters to the resource archetypes. Therefore, the clusters were evaluated using a more general comparison of the centroid and archetype values, applying a design-centred evaluation of the dynamics patterns each of the centroid attributes described.

This result was then evaluated by analysing the archetype attributes of the individual resources in each cluster, and determining if they generally matched the expected archetype of the cluster.

Comparing the cluster groups to the archetypes of the resources assigned to them, it was possible to measure their accuracy. The clusters categorized the resources with an average accuracy of 59%, 64%, and 69% for each of the 3 clustering analyses respectively.

The tables IV-VI show: The resource count of each cluster, the archetype it most resembles, the percentage of resources in the cluster that matched the cluster type, and the most significant groups of resources wrongly included in the cluster.

TABLE IV  
Results from clustering resources using 6 centroids

Resource count	Archetype	Accuracy	Significant errors
15	Skill Point	67%	27% Currency
15	Material	47%	26% Currency 13% Potion
28	Potion	40%	39% Ammo 11% Rage
17	Currency	71%	
17	Health	100%	
27	Mana	48%	22% Rage 15% Ammo

The first analysis with 6 clusters demonstrated the clusters starting to become condensed into definitions similar to the archetypes that were expected to exist. There were some expected archetypes still missing, and their associated resources were distributed amongst other clusters.

TABLE V  
Results from clustering resources using 9 centroids

Resource count	Archetype	Accuracy	Significant errors
14	Skill Point	71%	21% Currency
18	Material	39%	28% Currency 28% Potion
21	Potion	38%	48% Ammo
3	Action	100%	
12	Rage	75%	25% Ammo
21	Mana	62%	24% Health
16	Currency	75%	
14	Health	100%	

With 9 clusters, more archetypes such as Rage and Action appeared, as was expected.

TABLE VI  
Results from clustering resources using 12 centroids

Resource count	Archetype	Accuracy	Significant errors
9	Skill Point	78%	
9	Material	67%	
25	Potion	52%	40% Ammo
3	Action	100%	
11	Rage	73%	27% Ammo
22	Mana	59%	23% Health
16	Currency	56%	31% Skill
9	Health	100%	
1	Skill or Currency	(100%)	
5	Health	100%	
9	Currency	89%	

Increasing the number of clusters to 12 did not show any additional archetypes, but did create new clusters whose centroid attributes were similar to other clusters, when inspected from a game design perspective. This either indicated the existence of sub-archetypes that should have shown up in a thorough qualitative analysis, or an underlying difference in how automated and design oriented analysis weighted the resource attributes.

One outlier did appear that seemed to lie in between Skill Point and Currency, but this was most likely a consequence of the huge variety of resources that can be designed for games. In an analysis with significantly more data points, single outliers such as this one would probably not create outlying clusters of this kind.

Having removed the structure of the qualitative analysis during the data pre-processing, the cluster definitions were unable to replicate the archetype definitions exactly, but most of the clusters still had direct counterparts in the archetype list, in the sense that they shared a number of attributes and showed the same dynamics patterns.

Out of all the generated centroids, the ones labelled Skill Point and Health were the most similar to their respective archetypes as defined earlier. In fact, they appeared with the same centroid attributes in all 3 cluster analysis described. There are two reasons for these correlations:

1) *For Skill Point resources*, the reason was that the usage of skills across many games and genres is identical, making the cluster very accurate. Skill Points are unique in this aspect, as other archetypes often follow common trends

without sharing as many identical attributes.

2) *For Health*, the reason for the strong correlation was the existence of a specific attribute, named vital, which appeared in all of the resources of this archetype, but rarely showed up in resources of other archetypes. This, together with a few other recurring attributes, made this cluster appear very consistently.

The Health cluster was especially interesting because of the fact that all the members of the cluster also were correctly identified as belonging to the Health archetype.

Some of the archetypes defined in the qualitative analysis did not appear in the cluster analysis. This seemed to be the most significant difference between the two methods. The main discrepancy was a lack of a cluster representing the Ammo archetype. This archetype was very prominent in the qualitative analysis and had very distinct dynamics, so it was expected to show up as a separate cluster. However, these dynamics was centred on one specific property named purpose. Resources of the Potion and Ammo archetypes are similar in most aspects, including them being single purpose resources, producing diminishing effects. The main difference is that Potions enhance the player while Ammo inflicts damage on opponents. Of course, when a clustering algorithm makes distance calculations, all attributes are weighted equal, meaning that the difference in purpose is no more important than any other attribute. The result was that Ammo was clustered together with Potion, a grouping that was perfectly natural for an automated analysis, but would not make sense from a design perspective.

The remaining two archetypes that did not appear in the cluster analyses, Rune and Physical, were not as strongly defined in the qualitative analysis, and with only three resources belonging to those archetypes, this kind of result was expected. As with the outlier data, a bigger dataset might have produced a better and more consistent result.

The resources themselves had interesting distributions amongst the found clusters. In the first analysis with 6 clusters, the only 6 of the 11 archetypes could be represented in the form of clusters, so all resources of the unrepresented archetypes were distributed to the clusters whose centroids most resembled their archetype descriptions. This changed as the cluster number increased. For instance, when the number of clusters was increased to 9, a cluster resembling the Rage archetype appeared and all resources of this archetype, which previously had been divided amongst the Potion and Mana clusters, were relocated to this new cluster.

In the first two clustering analyses, Potion resources had been divided amongst the Potion and Material clusters, but in the 12 cluster analysis, all resources with the Potion archetype were finally gathered in the expected cluster. This was most likely because of a general refinement of the cluster areas as the cluster count increased.

There were a number of resources that were clustered differently than their archetype would indicate. This was



most likely a result of the different structure of properties in the resource and archetype descriptions used in the two methods.

## VII. DECISION TREE CLASSIFICATION

In order to test the consistency of the archetypes from the qualitative analysis, a Decision Tree classification analysis was done [12] using the J48 algorithm with the archetypes values of the game resources as the target value.

The resulting Decision Tree is shown below.

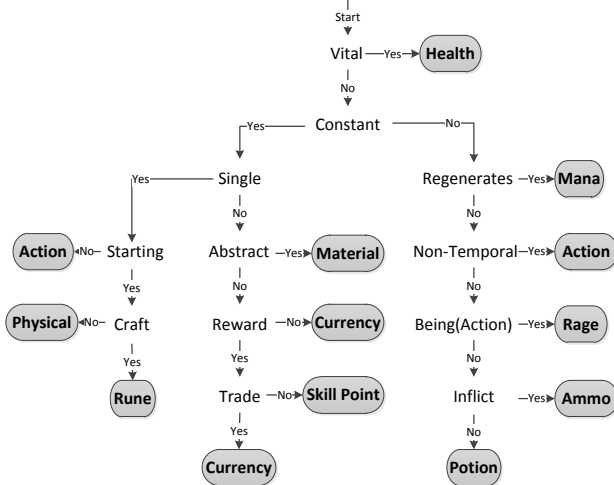


Fig. 3: J48 Decision Tree classification of archetypes.

The algorithm correctly predicted the archetypes 104 (87%) of the 119 resources used, using the Weka default settings. This was a very good result when comparing to a baseline ZeroR result of 19 (16%). The algorithm was run repeatedly with different settings, but they mostly produced the same result.

Since decision trees are based on describing the importance of each property in defining a target property, it appeared to be well suited to evaluate the results of the qualitative method used earlier. For instance, in the decision tree it is easy to see the difference between Ammo and Potion resources. The ability of Ammo to inflict damage or other negative effects to an enemy is its defining feature, and one that makes such a resource easily recognizable in any qualitative analysis. The decision tree seemed to accurately describe the relative importance of each attribute. In contrast, the clustering algorithm had problems differentiating between Ammo and Potions. This was due to the inflict attribute contributing just as much as the other 43 attributes to the distance function used. This important attribute simply lost its significance due to the weight of the others.

While the algorithm does not prescribe the categories that are to be found, decision tree classification seems like a good tool for verifying categories that already have been defined and giving feedback on bad groupings. For instance, the Currency archetype seemed to be loosely defined in the qualitative resource analysis, and the resources with this

archetype were a bit scattered in the cluster results. Combining those results with the fact that Currency appeared twice in the decision tree indicated that this specific archetype might have to be redefined.

The decision tree analysis produced a good visual representation that would be useful to game designers due to its simplicity. A chart like Fig. 3 conveys very well how the attributes are linked to each archetype and displays their relation to each other.

## VIII. FREQUENT PATTERNS

The Apriori frequent pattern algorithm was also used on the data set [12] to explore how common themes of certain resource attribute combinations had been used across different games. This process yielded 38 different patterns with a lift over 1.5 and 52 with a confidence of 1. These rules showed interesting recurring themes in resource attributes, which could be examined for patterns of game design and dynamics. The following example demonstrates how this information could be useful.

One of the rules found was: Storage cap = Unlimited ==> Purpose = Enhance & Consequence = Constant. (Confidence = 0.97, lift = 2.41).

In short, this rule indicated that resources with infinite storage are almost always usable for purposes that stay constant in the game and enhance the player. One example of a resource following this rule is minerals in StarCraft II.

Analysing the rule for its game design implications would indicate that designers wanted players to be able to make their own decisions on when to spend these resources and be able to collect any number of resources without being forced to spend them before reaching a storage cap. These kinds of resources are often used in build-up phases of a game, where players are encouraged to evaluate long term implications of each of the spending options for the resource.

## IX. DISCUSSION

The two different approaches to analysing game resources, one design centred and the other through machine learning each have advantages and disadvantages.

Using expert knowledge in a comparative analysis of game resources, it is possible to create a property framework that focuses on the essential components, describing them in sufficient detail for game designers and analysts to understand. Using this data, categories of resources can be defined, based on both the resource attributes lists and a design centred understanding of game mechanics.

The main disadvantage of this process is the very real possibility of bias in the categorization, leading to certain patterns being exaggerated while others go unnoticed. Also, there are limitations in the patterns that can be found by a qualitative analysis, due to the complexity of the resource and archetype descriptions, and the sheer amount of data.

The qualitative process of analysing game resources is highly dependent on expert knowledge and impossible to automate, but machine learning algorithms can be used as

supplements for the qualitative categorization process. When the resource dynamics have been systematized, well known data mining methods can be used to classify the data and recognize repeating patterns. Using these methods, it should be possible to limit human bias in the categorization process and to make more precise analyses of large data sets.

In order to properly evaluate structured data such as this, it should be possible to develop a recoding algorithm for the resource properties, before standard machine learning takes over. By doing improved pre-processing and recompiling of the data, it might be possible to avoid losing structure information in the conversion process. Alternatively, new machine learning algorithms could be developed that properly analyse the resource data without the need for interpretation, for example by custom distance functions in clustering algorithms. It is possible that such tools already exist, but we are not aware of them.

Whatever the method, the goal would be to input some human expert knowledge into the procedure, to ensure better automated analysis of game resources.

Human bias in this research is inevitable. The selection process of games, attributes, properties and definition of resource systems are all created in a qualitative manner. However, the area of resource systems is a new field, so no established tools or research was available, and the bias may be reduced over time through dissemination: By creating a vocabulary for discussing resource systems, we allow for future contributions to be made in the field by other analysts.

Any framework that is intended to be used for game design and analysis will have to use a descriptive vocabulary. This means that there are going to be semantics and implications for game dynamics that machine learning tools will be unable to recognize. This was clearly demonstrated in the example of clustering of Ammo and Potions above. Resources of these two archetypes are similar in many aspects, but their difference is obvious and would show up in any qualitative analysis. Automated processes have no way of recognizing these important differences.

## X. CONCLUSION

The procedure of structuring qualitative analysis of digital games and applying machine learning methods has been done before, but those efforts have mainly been centred on analysing player personalities and behaviour. This paper represents an attempt to diverge from these and use the approach on one specific aspect of game mechanics. The results are encouraging and would seem to open up the possibility for more research. This includes:

- 1) *More data analysis centred on properties and attributes*, including better visualizations.
- 2) *Developing a working syntax*. The current model defines the framework and vocabulary of game resources, but lacks the syntax necessary in creating an intrinsic language for formally discussing resources.
- 3) *Structured testing of the model as a design framework*.

For instance, the usability of the frequent patterns and decision trees could be tested in a design and analysis environment.

It is possible that similar methods could be used for other aspects of game design, such as designing level, puzzles, NPC's and user interfaces, but it is hard to predict how successful such efforts would be without substantial research being done on the subject.

There certainly are challenges to this approach, both in the qualitative analysis of games, the following automated analysis of the data, and merging the two. But if successful, it would lead to a more comprehensive vocabulary for analysing and discussing game mechanics, novel methods of designing games, and hopefully, better games.

## XI. REFERENCES

- [1] Tobias Mahlmann, Anders Drachen, Alessandro Canossa, Julian Togelius and Georgios N. Yannakakis (2010): Predicting Player Behavior in *Tomb Raider: Underworld*. Proceedings of the IEEE Conference on Computational Intelligence and Games (CIG), 178-185.
- [2] A. Drachen, A. Canossa, and G. N. Yannakakis, "Player Modeling using Self-Organization in *Tomb Raider: Underworld*," in *Proceedings of the IEEE Symposium on Computational Intelligence and Games*, Milan, September, 2009.
- [3] Andre Gagne, Magy Seif El-Nasr, and Chris Shaw. A Deeper Look at the use of Telemetry for Analysis of Player Behavior in RTS Games. *International Conference on Entertainment Computing (ICEC 2011)*, 2011.
- [4] Adam M. Smith, Chris Lewis, Kenneth Hullet, Gillian Smith, Anne Sullivan. An Inclusive Taxonomy of Player Modeling. Technical Report UCSC-SOE-11-13, 2011.
- [5] C. Thureau and C. Bauckhage. Analyzing the Evolution of Social Groups in *World of Warcraft*. *CIG 2010*.
- [6] C. Thureau, C. Bauckhage. Group Evolution Patterns in *World of Warcraft*. *GI Jahrestagung 2010*
- [7] Yee, N.; Ducheneaut, N.; Nelson, L.; Likarish, P. Introverted elves & conscientious gnomes: the expression of personality in *World of Warcraft*. *ACM CHI Conference on Human Factors in Computing Systems*; 2011 May 7-12; Vancouver, BC, Canada.
- [8] Staffan Bjork, Jussi Holopainen, *Patterns in Game Design (Game Development Series)*. Charles River Media Inc., Rockland, MA, USA. 2004
- [9] Hunicke, R., LeBlanc, M., Zubek, R. 2004. MDA: A Formal Approach to Game Design and Game Research. *Challenges in Game AI Workshop, AAAI-04*. AAAI Technical Report WS-04-04.
- [10] Peter H. Rasmussen, Tróndur Justinussen. 2012. Resource Systems in Games: An Analytical Approach. <http://itu.dk/people/tjus/Resource%20Systems%20in%20Games,%20An%20Analytical%20Approach.pdf>.
- [11] Espen Aarseth, Solveig Marie Smedstad, Lise Sunnanå. A multidimensional typology of games. *DIGRA Conf. 2003*.
- [12] Jiawei Han, Micheline Kamber *Data Mining: Concepts and Techniques*. Morgan Kaufmann. Burlington, MA, USA. 2006
- [13] Remco R. Bouckaert, Eibe Frank, Mark A. Hall, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. WEKA-experiences with a Java open-source project. *Journal of Machine Learning Research*, 11:2533-2541, 2010.