

# 琉球大学学術リポジトリ

## インターアクティブな文学作品インターフェースの構築

メタデータ	言語: 出版者: 琉球大学教育学部 公開日: 2007-07-18 キーワード (Ja): キーワード (En): 作成者: Taira, Katsuaki, 平良, 勝明 メールアドレス: 所属:
URL	<a href="http://hdl.handle.net/20.500.12000/1038">http://hdl.handle.net/20.500.12000/1038</a>

## Interactive Window on Literature

Katsuaki TAIRA

If there is such a thing as an interactive aid to literature it is going to advance the understanding of deep undercurrents that underlie literary pieces as well as that of the works themselves. The consideration becomes especially pertinent if we think of terse and compacted literary works like poetry. What about if the situation is further delimited, or perhaps more complicated, by the presence of another parameter: the interpreting agent who does not share the cultural ethos that is directly responsible for the production of the artworks in question, although the causational order here depicted may be modified if you want to be more accurate. What about if the integrated digital elements function as a kind of exegetic aid to comprehensively grasping the various factors that are involved in the target artwork? Would not such a scheme be worthwhile to pursue because it promises to allow such “alien” subject to delve further into what used to be encapsulated world of literature? Such simple yet ambitious hopes and questions are the jumping off points for me to dare making the excursion of the kind I am proposing here. To synergize the two realms of literature and digital media and combine and amplify each in order to bring out the best the former has to offer in terms of the experiential enrichment such subject above indicated can hope to attain. The work seems rather deceptively simple, I must say. It entails pragmatic/programmatic effort to hitch together the previously alien elements and somehow locate the points of intersection that are deemed to be rife with structuring possibilities to give rise to the synergy of the two. On the more fundamental level, the enterprise adumbrated becomes more of a methodological issue. How to integrate the two distinct and apparently disparate entities into a seamless whole that in the end yields something meaningful and transcends the comprehension traditionally enjoyed by the alien subject noted above? Obviously that in itself is a knotty issue not likely to be solved without many a Gordian hackings. At the inception of this essay I am resigned to that, in truth. I am willing to go through awkward forays that may or may not result in the aimed objectives. But when I think of the potential synergy that is awaiting to be realized between the two realms I am compelled to barge on, regardless of the difficulties that are predictably scattered throughout the journey. As for the more radical question of whether or not the two realms here mentioned really have the

areas that can successfully be dovetailed with each other, I will leave to the more philosophically minded students. That question is simply too inhibitive on a mind that is busily searching for new venues to merge seemingly heterogeneous genres in order to attain new levels of semio-semantic possibilities, even though that necessarily entails iconoclastic challenge to the traditional standard.

Now let us consider the following seemingly very simple quatrain written in the format of the ballad. Although underlying significatory layers can easily be conjured up without undue mental exertion (since even the folkloric formulation can easily be embedded in the rich tradition of a nation's cultural history), the seeming unsophistication of the ballad may pose a better challenge to the scheme we are trying to build according to the synergistic plan I have laid out above. Without much delay then I will quote the target lines below.

O what can ail thee, knight at arms,  
Alone and palely loitering?

The sedge has wither'd from the lake,

And no birds sing. (from "La Belle Dame sans Merci: A Ballad")

It is once again more or less a straightforward narration of a knight who is benighted possibly at the lake. But the tone is, not surprisingly, rather mysterious as if the author is inviting the reader to attach as many associative significations to the poem as possible<sup>1</sup>—even the kinds that are not directly proliferated by the printed words. How does one make of the poem then? That becomes a rather knotty issue. The more so, the more the reader is distanced from the cultural ethos that gave rise to the poem.<sup>2</sup>

---

<sup>1</sup> It may not be irrelevant to add in this context that Keats is prone to fragmentary inspirational outbursts by which he was more often than not successful in investing his work with the depths that are truly remarkable. In M.H. Abrams words, Keats exhibits "proclivity for fragments that come on one with a fine suddenness." That is one of the reasons why his poems are rife with recondite significatory jewels that are often beyond words. For more on the fragmentary inspirational tracks Keats writing takes, see M. H. Abrams, *The Mirror and the Lamp*, pp.136-137.

<sup>2</sup> Although it may not do much good to justify the inclusion of this particular poem as a candidate for synergization between literature and digital media, suffice it to remind you of the comment that redoubtable Romantic scholar M.H. Abrams made on the Romantic natural who died prematurely:

Keats . . . figures mainly insofar as he represented in some of his poems a central romantic subject: the growth and discipline of the poet's mind, conceived as a theodicy of the individual life (what Keats called "a system of Salvation") which begins and ends in our experience in this world (*Natural Supernaturalism*, p. 13).

Granted that it is a rather grandiloquent observation; nonetheless the encomium granted Keats by none other than the authority of Romantic literature should ease the minds of those skeptics who are inclined to side with Byron, for instance, as the best manifestation of individualism and the best candidate for such "Frankensteinian"

Understandably any contemporary who is removed from the time when the poem was actually produced is without a doubt situated at least some distance from the creative spirit that pervades the poem in question. That means everyone who was not at the helm of the creation when the poem came out of the poet's inspirational vessel (in this sense even the Keats who did not coexist with the creative spirit when the same poem came into being can be considered somehow alien to the self who did, strange may it sound) is doomed to strive toward the center of the poem because of the hiatus that exists between the present subject and the other that is expressed through the poem. That is not an easy task. Let alone if the subject does not share the remnant of the cultural background that maintains vertiginous link to the ethos that gave rise to the moment of the poetic effusion.<sup>3</sup> We need to take that into consideration when we are readying ourselves to construct the digital platform to build an experiential space to delve into the spirit of the ballad and the layered significations that surround it. The exegetic design we develop then will directly derive from that understanding. Now without further delay let us begin the work we have adumbrated.

What would be the best way to open up the presentation that contributes to the viewing subject's appreciation of the ballad? Without a doubt utilizing the best the combination of the text and the digital media has to offer. How about deploying sound clips to grab and stimulate the curiosity of the viewer? In order to accomplish that the following script may be necessary.

On prepareMovie

```
Sound(pSndCh).queue([#member: pIntroSndMem, #rateShift: pPtch, #loopCount:
pRptNm])
```

```
Sound(pSndCh).play()
```

End

Needless to say all the variables that are preceded by the p are to be declared and assigned values before the current handler runs. This is a simple sound initiation script. But what if you want to show a series of graphics or video in synch with the

---

amalgamation as literature and digital media.

<sup>3</sup> In fact what complicates interpretation of artistic works that are derived from different cultural context than the one where interpretive subject resides is that other than the aesthetic criteria used to evaluate the works in question there is such a thing as more subtle social conditions under which any work of art constitutes itself, and thus judged, in. As Ellen Schauber astutely notes in her critical book, Keats is particularly susceptible to this latter intangible aspect of social trend because of his tenacious adherence to the "truth beauty" tenet. On more about the subtle intangible value system mentioned here, see the argument by the same author at *The Bounds of Interpretation*, pp. 182-183.

music? That is likely to be a more promising approach to bring the viewer into the right psychological state to appreciate the text that is to come. In order to present a single graphic that remains on the monitor as the viewer waits for the end of the sound clip another simple coding suffices. The following is just an example of that.

```
Sprite(pGrphCh).puppet=1
```

```
Sprite(pGrphCh).loc=point(pH, pV)
```

```
Sprite(pGrphCh).member=member("pInitPrMem")
```

However, this may be a little too simplistic an approach. If the sound that is to be triggered with the inception of the presentation and is going to last some time, then a single static image may not be able to hold the viewer's attention for the whole duration. There needs a variation. One way to achieve it would be to animate, or rather present a series of images in a sequence. Although each image still constitutes a static single durational moment, the sequence of them will be enough to divert the viewing subject to immerse him in a state that is preliminary to the textual experience that is to follow. To accomplish the animational scheme there are a number of possibilities (as with any structural scheme planned in any programmatic environment). One is to embed a line that triggers the necessary action within the same script and another is to invoke a script that is encapsulated as an object. Either one of them has advantages and disadvantages at the same time. Option for one over the other depends on the convenience of the moment rather than anything else. It is a profound and at the same murky subject which we had better not go into details at the moment. Suppose we are to incorporate the function within the same script. It can be accomplished in the following manner.

```
if the ticks>pInitT+pWtT*60 then
  sprite(4).member=member("initGrph"&&pN)
  pN=pN+1
  if pN>pGrphIndxLmt then
    pN=1
  end if
  pInitT=the ticks
else
  exit
end if
```

Here the first variable pInitT represents the time that is recorded the moment the presentation starts. The next variable pWtT holds the value that is used to compare the difference between the pre-recorded time and the current time indicated by ticks.

Once the condition in the first if clause is satisfied then the lines subsequent to that are run. The pN concatenated to the string "initGrph" indicates the index number attached to each graphic member that is to be presented as the viewer watches the monitor prior to the textual introduction. Every time member assignment is invoked then the index is incremented to ready for next member assignment. However, the incrementing process cannot go on forever because there is only a limited number of graphic members to be prepared, or more precisely, there is no need to prepare more than necessary number of graphic members for the sake of preparing the viewing subject for the next stage. Thus when pN reaches the index number limit, here represented by pGrphIndxLmt, then pN is returned to 1, the very first graphic member. And each time the graphic member is changed the value represented by pInitT is initialized to the current time, allowing the next wait and replace cycle to start. In fact because of that wait time delimiter no member change takes place if the temporal condition is not met which is set in the first if clause. That is the summary of the setup to present a sequence of graphical elements in synch with the music that is playing. Another alternative may well worth introducing here. The seemingly animated display of graphics may be satisfactory to a certain extent in that it prepared the viewer for the next textual phase that is to come at the termination of the sound clip. However, nothing is more expressive and potentially compacted with layers of signficatory suggestiveness than a video. If the editor chooses to incorporate a moving picture into the target presentation then that is certainly a possibility. The setup is relatively easy. The simplest way to implement the scheme is to send the playback head to the section where the video actually plays until either the viewer decides to the next section or the playback head is automatically moved on to the next section at the termination of the video display. Once the playback head reaches the designated section the following script takes control of the video playback.

#Play:

videoplay(sprite the videoSprite of me)

#Pause:

videopause(sprite the videoSprite of me)

#Rewind:

videoseek(sprite the videoSprite of me, the segmentstart of sprite the videoSprite of me)

#StepForward:

videopause(sprite the videoSprite of me)

videoseek(sprite the videoSprite of me, the currenttime of sprite the videoSprite of me+30)

#StepBackward:

videopause(sprite the videoSprite of me)

videoseek(sprite the videoSprite of me, the currenttime of sprite the videoSprite of me-30)

#Seek:

videoseek(sprite the videoSprite of me, the param1 of me)

#PlaySegment:

videoplaysegment(sprite the videoSprite of me, the param1 of me, the param2 of me)

Although the coding is shown only partially, the general concept comprehended here should be rather obvious. As long as the presentation remains at the current segment, viewer is given the power to manipulate the presentation in a way any conventional means allow him to. Once the playback head reaches the video segment there are various possibilities to enhance the viewing experience of the interactive subject. For one thing he can enjoy a dialogue with the material shown right in front of his eyes. One method to accomplish it is to embed cue points in the digital video and when the right time comes as the video progresses along its temporal sequence, they trigger preprogrammed actions. Look at the following example.

on cuePassed me, chnnlNum, cueNum, cueName

if cueName contains "dltTx" then

member("videoF").text=""

end if

if cueName contains "grtns" then

sprite(25).member=member("videoF")

sprite(25).ink=36

member("videoF").text="This is something you will enjoy very much."

member("videoF").foreColor=53

set the foreColor of char 8 to 12 of field "videoF"=71

set the foreColor of char 22 to 30 of field "videoF"=14

end if

etc.

This should be telltale for some who have been following my argument. The handler preceded by *on cuePassed me* receives the signal those pre-embedded cues emit. Once they accept them, in other words cues match the string values enclosed in the quotation marks within the selfsame handler, then the action encoded in the lines within the if conditional clause is triggered. In the first if clause in the current example the videoF member content is vacated, i.e. any text the target member holds is deleted and as a result no visible element is left on the stage. By the same token when the cue value

coincides with the string value indicated within the quotation marks in the subsequent conditional clause, the lines enclosed within the if clause run. When that happens the member of channel 25 is set to videoF, the ink property of the channel (the transparency of the part that is apparently empty but within the perimeter of the theoretical rectilinear area of the object) to transparent, the text of the member that resides in the channel to the string value indicated within the quotation marks, the general color of the text to the color value indicated by the number, and those particular letters indicated by the following lines to the color values designated on the right side of the equations. This mechanism is very useful because it allows synchronization many actions. For instance, as I mentioned above, certain scenes can be set up to lead to another related actions except that those related actions do not materialize unless the interactive subject specifically invokes them by a demonstration of his intentions (clicking the mouse, pressing a certain key, and so on). This mechanism opens up a whole new world of opportunities for those who are seeking new ways to link the linear presentation of a video with some other related gamut of elements that have traditionally been related to the footnote status. They do remain somehow behind the scene, as it were, in that they do not present themselves unless the viewer specifically requires them, as I have noted above, but the fact that they do and can take over the main interfacial stage at a moment's notice indicates that the combination of the digital means with the textual, or more accurately all the ingredients emanating from the original textual, content indeed promises a powerful synergistic force that is out there waiting to be tapped by every possible modus available.

After the introductory ambient phase comes the text proper presentation. Now let us consider the optimal structure in which to combine the various digital media with the original text. As I mentioned above, the entire presentation is to be endowed with exegetic notes as well as exploratory forays into multi-layered significations to help the interactive viewer to better appreciate the literary text. With that confirmation in the comprehensive objective of our rather ambitious essay, let us delineate the actual nitty-gritty programmatic setup to make the goal even just a little closer to realization. Now focus on the first line, which reads, "O what can ail thee, knight at arms,/ Alone and palely loitering?" The most probable issue that arises when a subject who is not completely familiar with the style the author adopts here would be the syntax. Although the line is rather straightforward as far as poetic usage goes, it is not quite prosaic in a way any ordinary expository sentences are structured. How does one explicate and convey that gist to the audience who is staring at the scene of which the textual content is most likely to be a part of? There are innumerable possibilities.



Going over every single one of them would be the best strategy to arrive at the optimal candidate that will help the viewing subject better understand the text. But that is not practical when the space is limited. Therefore, the architectural demonstration that is to follow is quite a limited version and may be narrow in its scope. Nevertheless, an attempt to come to some sort of compromise in search of constructing a programmatic auxiliary to literary understanding and cultural appreciation of the other, I am sure, is worth in its object justifiable per se. If that point is agreed upon then let us plunge into the programmatic arcana redolent of the promised virtual space where potential synergy is almost perceptible.

Since I cited the syntactical issue that is most likely to arise in connection with the first line of the poem let us focus on the line and ponder if we can come up with the digitalized auxiliary that will help the uninitiated with better understanding of the layered meaning of the overall ballad. How about a mechanism that will read the target line and displays the prosaic interpretation, or rather rephrasing, of the line question? Look at the following coding.

```
on beginSprite me
  pFlag=0
  sound(1).queue([#member: member("ln1-2Bld")])
  sound(1).play()
end

on exitFrame me
  if pFlag=0 then
    if not soundBusy(1) then
      prTx="Why do you look so sick, dear wandering Knight? You look lonely and pale."4
      member(pPrPhrsMm).text=prTx
      sprite(pPrPhrSpNm).puppet=1
      sprite(pPrPhrSpNm).loc=pPrPhSpLc
      sprite(pPrPhrSpNm).ink=36
      sprite(pPrPhrSpNm).member=member(prPhrsMm)
      pFlag=1
```

---

<sup>4</sup> Paraphrasing is always tricky. This is meant to be just an example of turning the poetic line to a plane prose. I admit there are an unlimited number of possibilities to translate the given line. But what I would like to emphasize here is that paraphrasing and interpretation are inseparable. Once one enters the open-ended domain of subjectivizing the original artistic text, there is no clear demarcation between biased and non-biased analyses.

```

    end if
  end if
end

```

The first handler triggers the target sound clip, which should coincide with the display of the text (if that is preferred). Once the text is being read the next exitFrame handler keeps track of the soundplay condition. If the reading is over, which is represented by the conditional phrase `if not soundBusy(1)`, then the paraphrase text is generated in the channel designated by `pPrPhrSpNm`. The location of the text is set according to the parameter `pPrPhSpLc` and so is the ink (transparency) of the text field itself by the `ink` property. The final line of the coding sets the text to the string value defined in the first line of the current conditional clause. In the example shown here, however, the paraphrase remains on the scene as long as the viewer or the playback head remains in the current section. That may cause a problem when the reader wishes to concentrate on the original text. In order to solve that issue it is advisable to give the interactive subject control over the paraphrase text. How can it be accomplished, seeing that the member that appears through the conditional coding resides in a puppeted temporary channel? That should not be too difficult. One solution would be to create a switch that is simultaneously embedded within the visible member itself as it appears on the stage. Look at the following coding.

```

paraObj=new(script "paraPrt", pPrPhrSpNm)
goAppend(paraObj)

```

The above lines are to be inserted in the coding structure cited in the preceding example. This ensures that the text field that appears with the paraphrase is equipped with the switch that allows the viewer to control the state of the text, whether it is visible or apparently non-existent. As is obvious by now, the signal issued by the above lines generate an object. It is actually this object that gives the viewer control over the exegetic text. The object does not have to be complicated. The following lists the barebone structure of the object, or to be more accurate a script that will generate an object, that will meet the demands of the subject who wants to have control over the paraphrasal text.

```

on new me, prPhrSpNm
  pPrPhrSpNm=prPhrSpNm
  return me
end

```

```
on goAppend me
  if (sprite(pPrPhrSpNm).scriptInstanceList).getOne(me)=0 then
    (sprite(pPrPhrSpNm).scriptInstanceList).append(me)
  end if
end
```

```
on rightMouseUp me
  sprite(pPrPhrSpNm).puppet=0
  (sprite(pPrPhrSpNm).scriptInstanceList).deleteOne(me)
end
```

The structure of this object generating script should not baffle us. The first handler preceded by `on new me` generates the object itself when it is invoked by the lines in the previous script. The argument sent over from the calling script is stored in the variable `pPrPhrSpNm`. The value now made available by the declaration of the property by the same name is used repeatedly as the channel number is repeatedly used to attach a new script to when it has not yet been associated with the channel (`on goAppend me`) or to turn it off when the viewer decides he has seen enough of the exegetic interpolation (`on rightMouseUp me`). This is a brief introduction to the parent script that is utilized to generate an object. It suffices the need to turn off the puppeted channel to go on to the next phase of the presentation. However, if the viewer decides to read the paraphrasal line then he is left stranded without any recourse to the mechanism to give rise to the same popup exegesis. We need something else. Suppose we agree upon a setup in which the original text field acts as a switch to turn on the explanatory string field. That may or may not be the best solution to the mild quandary we find ourselves in. Suppose, however, that is what we wish to accomplish. Then how can that programmatic structure be realized within the synergistic environment we are attempting to create? The answer is quite simple. It is found by placing the similar structure within the original text that is already a part of the structure of the original script attached to it. If you remember the initial exegetic component is triggered by the Boolean value that is dependent on the sound (to be more accurate, whether the sound is playing or not) triggered at the inception of the presentation. If the conditional clause comes up with the positive answer the popup text field comes into view in conjunction with the object that controls the state of the channel used to displays the exegetic text. The similar architectural design can be

repeated to generate the popup filed except for a minor alteration. This is time the triggering agent is purely volitional. The realization of the same filed is contingent upon the will of the viewing subject. Therefore, there needs to be a mechanical signalization initiated by the subject that gives rise to the situation that previously occurred automatically. In the following instance, the intercession is made by the handler on mouseUp.

on mouseUP me

prTx="Why do you look so sick, dear wandering Knight? You look lonely and pale."

member(pPrPhrsMm).text=prTx

sprite(pPrPhrSpNm).puppet=1

sprite(pPrPhrSpNm).loc=pPrPhSpLc

sprite(pPrPhrSpNm).ink=36

sprite(pPrPhrSpNm).member=member(pPrPhrsMm)

paraObj=new(script "paraPrt", pPrPhrSpNm)

goAppend(paraObj)

pFlag=1

end

This may suffice the need to turn on the same exegetic entry as many times as the viewer wishes to. But what if one wants to enrich the exegetic entry and randomize the string value that is initialized to the target text field every time the embedded switch is turned on? Although randomization cannot go on indefinitely because of the limitation (practically speaking, that is) on the number of string entries that will be used to fill in the text filed, the randomizing scheme would certainly broaden the horizon the viewer acquires vis-à-vis the target lines as well as the whole ballad. After all he will be brought in contact with the multifarious significatory phases that may be hidden from cursory glances. By making at least some of those layers explicit, the viewing subject is expected to have better chances to broach the surface resistance such sentences may pause. Now without going much further away from the proposed randomization, let us look at the actual coding that helps to accomplish the scheme we have been adumbrating.

on beginSprite me

prphrsList=["Why do you look so sick, dear wandering Knight? You look lonely and pale.", "A lonely knight wanders through the wilderness. He for some reason is unwell. The mystery is that what a knight like him is doing there.", "You are encouraged to think why the knight is wandering in a lonely setup. What is he in

```

search of, if there is any aim at all?"]
end
on exitFrame me
  if pFlag=0 then
    if not soundBusy(1) then
      prTx=prphrsList.getAt(random(prphrsList.count))
      member(pPrPhrsMm).text=prTx
    end if
  end if
end
on mouseUP me
  prTx=prphrsList.getAt(random(prphrsList.count))
  member(pPrPhrsMm).text=prTx
end

```

This is the amended structure to enable the randomization of paraphrases. As has already been noted, the basis structure remains intact. The script itself consists of three handlers as has been the case in the previous example. What is different is that under the randomization scheme a list is set up in the beginSprite handler to generate the target paraphrase out of the whole number of the entry enclosed in the list. The more entries are enclosed in the prphrsList, the more unpredictable becomes the next entry when the paraphrasal text field is reinitialized. Other than that the two examples work identically. No matter how many exegetic entries are enclosed in the list, the rest of the operation runs without a hitch because the program comes up with the number of the entries in the list (prphrsList.count) and picks up a randomized paraphrasal candidate in the manner specified as prphrsList.getAt(random(prphrsList.count)). Repopulating the text field with the new string value is essentially the same as in the fixed text scheme explained above.

Let us hurry on to the next lines and see if we can elucidate on them with the synergistic means as in the previous examples. First let us remind ourselves the actual lines: "The sedge has wither'd from the lake,/ And no birds sing." Admittedly this is a rather prosaic line. But in its simplicity lies recondite semantic layers that may not be obvious to the uninitiated. Whether it is actually rife with cultural and linguistic significations or not let us dissect and translate it into something truly exegetic. First we need to skim the surface and salvage the gist that floats from the semiotic interplay each word is being engaged in. "The sedge that used to grow around the lake has died. The birds are no longer seen around the locale." That is truly

mundane but at least the signifiatory hold the interactive subject establishes at this stage is essential for the further appreciation of the lines. Once the obvious comprehension of the target lines is made then other perspectival attempt to delve into the semantic layers of the ballad gets underway. The sedge has withered because the cold whether advanced to the lake area. With the cold birds that used to carouse over the lake fled south or at least sought shelter in the warmth of the forest. A part of the interpretation is a mere conjecture but not entirely whimsical. Associating the drying out of the sedge with the seasonal change, for instance, is more congruent with the mood the first two lines have established. Although there may be cited various contingencies that would induce the conditions that are described in the current lines, imagistic context that can more readily be aligned with the rest of the ballad should certainly be a strong candidate to be inserted in the exegetic list that will pop up as the viewer peruses the relevant text field in connection with the two target lines. Now how about giving additional names for the sedge if that is deemed to help the viewing subject better comprehend the lines? How about incorporating more interpretational conjectures if that will broaden the viewing subject's cognitive horizons? Pursuing the same spirit I will also add the following paraphrasal insertion:

The sentiment the withered sedge evokes is something lonesome and lugubrious. The second line corroborates the desolate atmosphere by the reiteration of the idea of desertedness by the disappearance of the birds. The aim of the knight's errands, which is said to be undefined, further emphasizes the mood of desolation and aimlessness, although the latter ethos also hints at the mystery that is embedded in the whole ballad.

Admittedly the above is a rather lengthy insertion if it is going to appear in the text field. But all the examples shown so far just indicate the wide ranging possibilities such paraphrasal cues allow the viewing subjects to enhance the deeper understanding of the implicit significations that are packed within a mere couple of lines out of the entire poem. Let us, for the argument's sake cite the list that, under the given structural frame, will be used to hold the string values that will be accessed to elicit the textual candidate. The list itself should by now be almost self-evidently clear. All you have to do is to enclose the string values within the brackets and separate each entry by a comma. Or more technically speaking, simply replace the old content of the list in the script summarized above with the new ones just outlined. However, in order to make the impression that will be created by the insertion of the exegetic phrases into the text field more vivid and meaningful, how about triggering vocal recitation in conjunction with the appearance of the new text in the target field? The effectiveness

of the method, of course, needs to be tested extensively before actual presentation is implemented. Bu the essential structure of the relevant scripts should look as follows.

on mouseUP me

```
prTx=prphrsList.getAt(random(prphrsList.count))
member(pPrPhrsMm).text=prTx
sprite(pPrPhrSpNm).puppet=1
sprite(pPrPhrSpNm).loc=pPrPhSpLc
sprite(pPrPhrSpNm).ink=36
sprite(pPrPhrSpNm).member=member(pPrPhrsMm)
paraObj=new(script "paraPrnt", pPrPhrSpNm)
goAppend(paraObj)
pFlag=1
```

end

on new me, prPhrSpNm, kyWrdOne, kyWrdTwo, kyWrdthr

```
pKyWrdOne=kyWrdOne
pKyWrdTwo=kyWrdTwo
pKyWrdThr=kyWrdthr
if sprite(pPrPhrSpNm).member.text contains pKyWrdOne then
    sound(1).cue([#member: member(pKyWrdOne&&"rc")])
    sound(1).play0
else if sprite(pPrPhrSpNm).member.text contains pKyWrdTwo then
    sound(1).cue([#member: member(pKyWrdTwo&&"rc")])
    sound(1).play0
else if sprite(pPrPhrSpNm).member.text contains pKyWrdThr then
    sound(1).cue([#member: member(pKyWrdThr&&"rc")])
    sound(1).play0
end if
pPrPhrSpNm=prPhrSpNm
return me
```

end

These are, as indicated above, actually two scripts extracted from their main body. They might deceptively more complicated than they really are. That means some explications are due. The first handler functions as a switch to turn on the cue filed and once the field is on to reinsert other paraphrases that are related to the main text

that is being read or having been read. As is self-evident from the handler name, it is written to respond to the message that is generated when the mouse is clicked (or, to be more accurate, when the mouse button is pressed and then released). What the first line accomplishes is to initialize the variable to the value that is obtained by the operation described on the right hand term of the equation. What the right hand term does should not be so befuddling simply because it is written in dot syntax. In plane language the `getAt( )` function selects the candidate within the cue phrase list contingent upon the index that is determined by the value ultimately derived from the `random( )` function, which randomizes the integer value of the number of elements that is contained in the list named `prphrsLst`. The rest of the process is identical to the one we have covered above. But what distinguishes the current coding from the previous one is that the calling lines expressed as

```
paraObj=new(script "paraPrnt", pPrPhrSpNm)
goAppend(paraObj)
```

invoke the same script yet slightly with a different mechanism. When the handler within the parent script is called into action to create an object necessary to start the recitation that corresponds to the phrase shown in the target field, the lines implanted in the initialization segment simultaneously compares the paraphrasal text with the keyword sent over from the calling script. If the keyword is included in the target string object then the appropriate sound file is loaded and played back as the viewer watches the textual scene deploy before his eyes. The member names invoked by concatenating the keywords and the `rc` is meant to facilitate the seamless embedding of constituent members that are to play roles in the entire program. In fact they could be anything but maintaining the constant suffix in conjunction with the variant yet consistent naming should not only make the editing process simple but endow the whole presentation with the kind of organic congruency necessary for a truly successful program. Be that as it may let us hurry on to the next lines. Staying with the present associative issue between the text and the digital media supplementation may be profitable in that it might give rise to more fecund ground for creating the synergy between the two by bringing in more auxiliary elements into the picture and yanking the original poem and the latter in multifaceted formulations. However, time and space is limited. Rather than running the risk of becoming bogged down with the ever complicating issue of begetting the truly seamless and synergistic approaching to comprehending the two traditionally considered disparate realms, we have far better chances of leaving the attempt as it is in an half baked state and go on to the next lines and ratiocinate innovative means to coagulating the text and the digital media.



Now let me cite the target lines before starting on any plans that may be strategically deployed to enhance the reading experience of the interactively placed subject.

O what can ail thee, knight at arms,  
So haggard and so woe-begone?  
The squirrel's granary is full,  
And the harvest's done.

The first line is a verbatim echo of the corresponding line in the previous quatrain. That may not be surprising considering that the poem belongs to the category called ballad, as is amply suggested by the title. What becomes moot here is that whether the exact wording of the line merits different rendition in terms of its partial function vis-à-vis the comprehensive effects the entire ballad is assumed to acquire as a result of the schema in which the textual and digital media are dovetailed into each other. The issue is to be carefully deliberated and pursued to the final resolution even if the derivative problems multiply in the process. However, as I have been reiterating time and again throughout this essay, there is a dearth of time and space even if I am willing to be engaged in that potentially complicating enterprise. Let us simply be agreed upon the simple hypothesis that some variation on the former approach to render the target line is appropriate and perhaps desirable at this juncture. With that premise I will undertake further fusion between traditionally considered discrepant realms of text and digital media.

Because the target line is the exact reiteration of the previous introductory line, the effects that will be associated with it will be more consistent and artistically appropriate if they are to be somehow developed on some of those that are linked to the previous one. Let us recall that the methods we deployed in conjunction with the same line are those that utilized sounds and graphics. The current phase seems to warrant a similar content deployment. Without further delay let us construct the programmatic structure that satisfies the literary needs of the interactive subjects. Look at the following construct.

on exitFrame me

```
if pAniCh<pLstAniCh+1 then
    puppetSprite pAniCh, 1
    memNm=getAt(pMemList, random(pMemList.count))
    sprite(pAniCh).member=member(memNm)
    sprite(pAniCh).loc=pInitLoc
    sprite(pAniCh).ink=36
```

```

aniObj=new(script "circInnrPrnt", pAniCh, pInitLoc, pAniSndNm)
goAppend(aniObj)
pAniCh=pAniCh+1
go the frame
else
  go next
end if
end

```

This is intended to set graphic objects (or for that matter any visible elements) turning in concentric circles as they are incessantly created until the numeric conditions are met as stated towards the end of the coding. The handler `beginSprite` indicates that the all the lines embedded in the current coding are to be run as soon as the presentation gets under way. What actually controls the visible objects is the invoked script "`circInnrPrnt`"; however, in order for the general reader to follow the train of thought better let me explain what each line is intended to accomplish. In case the following explanation comes too abruptly I must add that the circular move that is engendered by the use of the series of functions is a visual metaphor and reminder of the reiterative strategy the poet employs in his pseudo-ballad. As the lines quoted above echo each other with only a few lines apart, the visual representation attempted here tries to mirror the conceptual gamut the author of the current work tires to compact into his inspirational offspring. Now let us go back to the actual script. The comparison between the two variables that hold integer values in the first line is intended to go along with the last comparative operation, which on the surface appears as an equation. What they do is that the animation channels represented by `pAniCh` are initialized into incremental integer values until the condition, that is `pAniCh` is more than the value represented by `pLstAniCh`, is satisfied. Once that happens the conditional lines sandwiched between the `if` phrases is skipped and no more circulating object is created. But while the comparison `pAniCh < pLstAniCh+1` is true the integer value represented by `pAniCh` is incremented by 1 every time the series of lines is run, giving rise to the channels populated by the visual images. What actually drives the circulating objects are the script that is invoked upon the `new( )` line and the one immediately proceeds from it. When the parent script is called a few arguments are sent to the destination script in order for it to determine the channel each discreet object is to inhabit, the location it is to set itself to vertically and horizontally, and the sound that is to accompany or to be associated with the given object, which will be used

to elucidate the target text in question. Once the encapsulated entity is created by the calling line, it is used to trigger the animation by the next line, which references by the calling phrase that is identical to the one actually being invoked. Now that is the bare bone mechanism that is compacted into the above script. Let us hurry on and study the script that is the main active agent in the cycle of the circulating objects in the digital-textual amalgamation. First the parent script itself.

```
--parent script "circInnrPrnt"
```

```
property pStep
```

```
property pAmount
```

```
property pOrigLoc
```

```
property pSpriteNum
```

```
property pFrequency
```

```
property pSpeed
```

```
property pOrigStep
```

```
property pAniSndNm
```

```
on new me, spNum, origLoc, aniSndNm
```

```
    pSpriteNum=spNum
```

```
    pOrigLoc=origLoc
```

```
    pStep=1
```

```
    pOrigStep=pStep
```

```
--try random value  between 60 to 100
```

```
    pAmount=random(160)+60
```

```
    pFrequency=8
```

```
    pSpeed=random(8)+2
```

```
    pAniSndNm=aniSndNm
```

```
    return me
```

```
end
```

```
on goAppend me
```

```
    if getOne(the actorList, me)=0 then
```

```
        append(the actorList, me)
```

```
    end if
```

```
    if getOne(sprite(pSpriteNum).scriptInstanceList, me)=0 then
```

```
        append(sprite(pSpriteNum).scriptInstanceList, me)
```

```
    end if
```

end

on stepFrame me

if pStep>=360 then

pStep=pStep-360

else if pStep<0 then

pStep=pStep+360

end if

angle=2.0\*PI()\*pStep/360.0

x=cos(angle)\*pAmount

y=sin(angle)\*pAmount

sprite(pSpriteNum).loc=point(x,y)+pOrigLoc

allowAnimation(me)

end

As you can see it is a rather lengthy script that in its entirety stretches from one end of the page to another (in fact a little more than that). It would be best if I just take a cursory look at the script, considering that the main objective of this essay is to accomplish the synergistic fusion between the two media and not elaborately analyze each constituent script, and proceed to the next segment of the original text. But because this particular script is so important both in its typical function and the manner in which it actually corresponds with the other types of script, such as the one that invokes it, I deem it essential to cover the inner structure and the implicit mechanism generated through the interlocked design of the whole. Now that is enough of a preamble. Let us focus on the target script for the moment.

If you by any chance wonder why there are so many property variables, which in fact functions as parameters throughout this script, they are there because a reasonable amount of variables tend to facilitate easy adaptation of the script. Once any contingency occurs, for example, it can be easily accommodated without modifying the entire script in any drastic manner. In this instance, the first variable is used to store the integer value that is needed to determine the rate at which the given object circulates around the given center point. In the above example the value is set to 1, the most detailed integer value without taking a decimal point. The second variable holds the value that enables the object to keep an orbital path. The third variable corresponds to the value sent over from the calling script and it is initialized to the points, expressed as integer values of horizontal and vertical coordinates, that are used to moor each object

to a fixed central location as it starts on an orbital journey. The next variable of course holds the integer value that is used to populate a channel used in the program. The variable pFrequency is a rate at which the circulating object halts before restarting its orbital move. However, this particular variable is left irrelevant at the moment because the script we are building now needs to chime with the idea of recurrence the reiterating line foregrounds as it brings the interactive reader back to the context in which he hopefully found himself previously. The rest of the variables need not be explained in any depth because they are self-evident by their given names except that the pOrigStep which is used to turn the object around the fixed center I referred to above. Now let us delve deeper into the arcane world of the parent script in order to bring the hidden mechanism out into the open, although they may not be so well-covered to the experienced eyes.

The handler that answers the call from the invoking script is the by now familiar on new me. In this instance, it is accompanied with a number of parameters. They are there to hold the arguments sent over from the invoking script, as I mentioned previously. The order of the parameters are important while the naming in fact is not in that no matter how they are termed they receive the corresponding values that are attached to the initiating code. Therefore, spNum, origLoc, etc. could assume any other names and still perform exactly the same functions as long as the terms used in the target script are consistent. However, naming is important in another sense because arbitrariness in naming parameters not only possibly confuses the editor as he tries to utilize the contents that are held within those containers called variables but also potentially breaks the organic consistency throughout the program that is necessary to fuse the whole entity into one seamless totality. As Shakespeare asks in one of his plays, What's in a name?, rose is a rose but does it smell and feel the same if it is called otherwise? Rather, is calling one thing spNum and other origLoc damaging their theirness and annulling the functions they are designed to be invested in? That is rather a philosophical question but at the same time a valid one too. There is potentially a number of ramifications. However, the most obvious and practical one would be that the semiotic significance one is accustomed to attach to one and the other is thwarted and the conventionally held semantic associations, similar in nature (I admit I sound terribly old-fashioned) to those propounded by Saussure generations ago, are put on a very precarious ground. So much for naming for the moment. The first line initializes the variable pSpriteNum to the value contained in spNum that is handed over from the calling script. The second line passes the value momentarily stored in origLoc to the property variable that is to be used throughout the script. The flowing

lines do more less the same thing by passing the temporary values to the property variables that are preceded by "p." The final return command returns the object created to the calling code as well as to the program itself so that the object generated becomes available from that particular scriptural vantage point. Now once the object is created it is used to trigger an animation and the consequent series of moves specified by the following lines in the same parent script. The triggering of the action concurs with the invocation of the goAppend handler. The first conditional line checks if the object just generated and currently used is if indeed already included in the program-embedded list named actorList. If not, which is expressed by the same equation with the 0 (zero) on the right of the expression (meaning false), then the current object is immediately included in the same list. That is enough to generate the kind of animation we are dealing with. However, another conditional clause is included in this particular script in order to accommodate for the various cued actions that may be useful to spice up the animation. These lines checks whether the object me is already included in the scriptInstanceList. If not, it is put in the list and used as a fulcrum to generate various interactive actions that are specifically linked to the target channel, here represented as pSpriteNum.

Because the actorList is a global list, meaning accessible from any place in the program, the designed animation continually occurs while other actions take place at various points locally. Once the object is included I the actorList then what happens is that a signal named stepFrme is emitted every time the playback head passes through that conceptual mark. The handler by the same name thus exists to intercept that very same signal. What that entails is that the coding set within the said handler is deployed every time the playback head goes through that conceptual construct and the message is emitted and intercepted by the on stepFrame handler. Now we will proceed and look at what is in store fro the target object as it is taken over by the commanding lines embedded in the current handler. Initially pStep variable is checked if the value held by it is equal to or larger than 360 degrees, which equals to one circulation of the object around the fixed point mentioned above or it is less than 0 (zero). If the former is the case then the value contained in the variable is set to the value that is arrived at by  $(pStep - 360)$  and if the latter is the case the same value is set to the value arrived at by adding the same to 360. In other words pStep is kept within the simplest value possible in order to keep the object circulating around the fixed point. Once the pStep is reduced to that optimum integer value then an angular value is obtained by embedding it as a numerator into the next line. The value obtained here is utilized to actually turning the object. The method used here is to multiply the ever changing values

generated by both `cos()` and `sin()` functions and plot the horizontal and vertical points of the object on a two-dimensional monitor/stage with the variable `pOrigLoc` supplying the fixed central point. The very last line `allowAnimation(me)` is meant to give the object halt-restart movements if controlled properly by the randomized value inserted in the invoked handler. However, since the current design does not require such stuttering motion we will ignore this particular coding for the sake of brevity in elucidating the overall scheme of this text-digital complementary amalgamation.

Needless to say the orbiting objects do not move around the respective fixed centers forever. There comes a time when the graphic/orbiting elements come to a stop and give way to the next phase of the textual presentation. The method that is utilized for the transition between the two phases does not have to be complicated. One would be an abrupt switch between the two segments by the `go` command, which moves the playback head to the next cue point. The other would be a transition accompanied by an effect. Although transitional effects triggered by a cue emitted by the moving playback head is not a fail proof method to enrich the current presentation, they could spice up the content by visually stimulating the interfacing subject with momentary transmogrifying images, either abstract or concrete. Among the various transitional types some may be more suited for the occasion than others. The following is just a sampler that at least hints at the possible interplays the said effect could involve the subject in.

`puppetTransition 2, 2, pTrnsTp, 0`

Here the variable `pTrnsTp` takes the value within the range of 1 and 52. Although I suggested using more or less situation adapted transitional types, the range is so great that each one needs to be selected with trial and error. While one seems to work well under given circumstances, the same effect may not under other situations, or even under the same situation except for a number of seemingly insignificant parameters. Therefore, the variable here given the name `pTrsnTp` needs to be filled in with a value with due consideration. Once the scene moves to the actual text-digital media amalgamation, a sound element may be an appropriate addition that constructively embellishes the interfacial presentation for the benefit of the activated mind. Therefore, the following lines are to be easily allotted in the current phase of the scheme to pull the charged mind into the poetic world that has already been panning out before his eyes for some time now.

`sound(1).queue([#member:member("hrkBckThm")])`

`sound(1).play0`

The trigger timing for the given sound element can be set to various points as the

playback head moves across the imaginary timeline. One way to release the sound would be when the text begins to appear on the screen. Another would be to have the sound or a set of sound clips ready to be played out upon demand from the interactive subject. While the text remains on the monitor some exegetic switches may be due in order to make the interface as interesting to the interactive subject as possible. A very simple mechanism is to create a channel as the subject moves the cursor point over the target text field and populate it with an appropriate string/text element. Needless to say the latter element has to be the kind that is deemed to function as an auxiliary signficatory amplifier as the inquiring subject tries to decipher the already materialized original text.

Let us now concentrate on the string values that will fill in the spot needed to create that very interface to facilitate the subject's appreciation and interpretation of the text. Although the exact lines have been quoted above, it may be worth reiterating to jog our failing memories at this point.

O what can ail thee, knight at arms,  
So haggard and so woe-begone?

One of the questions to be asked here, especially after the long ratiocination on the reuse of the leitmotif at the beginning of the target lines, is how and what to supply the exegetic auxiliary string chunk to give depth to the work as well as seamlessly transition it to the next line, which is not a repeat. Does the identical interpretation suffice with a possible linkage that organically brings the reading subject to the next dramatic stage that with its poetic subtleties saturate the first line with every shade of semantic nuances? Or a somewhat completely different tack should be taken in order to thrust the reader to a new conceptual landscape that is without a doubt also saturated with poetic nuances? Perhaps both may be true to an extent. The best option in this case would be give the subject the choice that allows him the former, more continuous interpretational approach while randomly turning on and off the latter and other angles as the program sees fit in its interaction with the hardware (as it were)<sup>5</sup>. Now let us proceed to the actual interpretation of the target lines given immediately above. Seizing upon the obvious semantic significance I have attempted the following interpretation.

What is troubling you, dear knight? You look so worn out and troubled?

---

<sup>5</sup> I may have phrased the schematic explanting rather vaguely and possibly erroneously. Each option the subject is given at any given moment appears exclusively in the text field provided on the monitor. The string options stored in the list tucked away in the depth of the program never appear simultaneously. If they did, it would be confusing to the subject, to say the least, and detract them from the main text.



That is rather simplistically put. How about the following one?

In a conversational tone the invisible I asks a knight, who is wearing a heavy armor and obviously in distress, "What is awry, sir?" The invisible I could not help himself from accosting the stranger, who is, the I is almost certain, mentally and physically tormented?

The preceding is a rather lengthy interpretation, I admit. But it adds significatory nuances to the original text that escaped from the previous one. If closer analysis of the text is in need the following may be more valuable in that it attempts to give the ballad multi-angled exegetic significations.

The knight is ailing obviously. The reason for which is not clear but judging from the repeated use of the first line the state in which the knight is presented to the audience has much more to do with the inner sufferings and the vicissitudes leading to them than the mere distress that is to be treated as temporary. The point may be reemphasized by the physical evidence that is textualized in the second line of the current couplet.

And so on. Possibilities are unlimited. Rather than cranking out exegetic candidates that are to be inserted into the program, let us actually look at the preceding string values being inserted into the overall design of the text-digital media amalgamation.

The method here deployed is essentially the same as used before. First put the candidates into a list.

```
exgLst=[frstStrCndt, scndStrCndt, ThrdStrCndt]
```

Then populate one of them in the text filed that will be created as the interactive subject moves his mouse over the target text. Needless to say, the mouseOver move can be replaced with the mouseUP move, as the editor sees it fit.

```
on mouseEnter me
```

```
    cursor 280
```

```
    sprite(pExgCh).puppet=1
```

```
    sprite(pExgCh).loc=point(exgH,exgV)
```

```
    sprite(pExgCh).member=member("exgTxFld")
```

```
    sprite(pExgCh).ink=36
```

```
    crntExgTx=exgLst.getOne(random(exgLst.count))
```

```
    member("exgTxFld").text=crntExgTx
```

```
    member("exgTxFld").font="arial"
```

```
    member("exgTxFld").fontSize=20
```

```
    member("exgTxFld").color=rgb(20,50,100)
```

```
    member("exgTxFld").fontStyle=[#plain]
```

end

on mouseLeave me

cursor -1

sprite(pExgCh).puppet=0

end

Many of the parameters given above are to be changed depending on the actual presentation that is deployed before the interactive subject's eyes. Such properties as font, fontSize, color and fontStyle are thus to be decided upon vis-à-vis the other parameters that constitute the whole scene. Without further delving into the minute mechanism of the above script let us hurry on and look at the original text. After all the original text is the fulcrum on which the contemplated amalgamation turns. Let me cite the following lines as a reminder.

The squirrel's granary is full,

And the harvest's done.

What do we make of this, first of all? Are they to be taken as a descriptive passage that tells the reader the time of the year? Or are they to be taken not only that but also a passage full of significatory semes to be deciphered in conjunction with the state of the knight's mind? Probably both. Now let us seize upon the first line and attempt an in-depth interpretation. When we encounter the line the first interpretive orientation most readers take would be seasonal. That is, when squirrels engage in that very act of storing away their food it is usually before the onset of the cold winter. Therefore, the most reasonable interpretation one could make in this context is diurnal. The same strategy to splice the poem into the diurnal cycle of nature sets the whole ballad against the natural cycle of floral and faunal backdrops. But what comes out from that initial reading is the contrast between the knight, including what his inner and outer self represent, and the entire universe that surrounds him. While the normal activities, represented by the filling of the granary by the squirrels, continue in the natural world, there is obviously something awry about the knight-errant, who is "haggard and so woe-begone." Without going extremes trying to account for the ramifying effects this contrast has on the whole poem, especially in conjunction with the knight's inner psychoscape, it is rather reasonable to conclude that the lines just commented upon are indeed a reflection of the knight's inner and outer state that will eventually redound upon the target heroin in the ensuing part of the ballad, and thus recursively throughout the poem. In fact, it is instructive to go on to the next segment and see what is actually taking place. Note the following description.

I see a lily on thy brow  
 With anguish moist and fever dew,  
 And on thy cheeks a fading rose  
 Fast withereth too.

As is immediately evident, as well as has been expected, the knight is now yoked together with the flowers, albeit in a rather metaphoric sense.<sup>6</sup> Regardless of the discrepancy that exists between the two realms that are spliced into one descriptive passage, what is organically accomplished in the present operation is that the parts of the knight's body and those of nature are yanked together in such a way that the linkage foregrounds the contrast rather than fusion between the two entities. In fact note how the lily and the rose in this passage are used in the diametrically opposed manner to the traditional metaphoric literary usage. During the Renaissance, for instance, it is quite common to describe the beauty of the lady's facial features by bringing in the organic associations these flowers had been invested with. But note how this time honored tradition is subtly reversed and the two flowers are thrown in to foreground the aberrancy of the knight's state, both physical and psychological. This strategy, however, does not function merely to emphasize the knight's condition in conjunction with the surrounding lugubrious natural landscape. The traditional element that is concomitant in the depiction of the knight so far is not, it turns out, completely discarded. That becomes evident when we move onto the next segment of the ballad. What is introduced in the very next line is a lady, after whom the title is named, who, according to the knight's narrative, which begins at this point, enchanted him and left him love sick and thus in desolation. What is important is, if I may hearken back to the traditional tropical usages of the flowers, is that they not only enhances the "tragic" lovelorn state of the knight-errant but also seamlessly assert themselves in that they become the traditional symbol of beauty and enchantment with the introduction of the lady in question.

It may be time that I cogitated upon the best interfacial structure in which to incorporate the crisscrossing of the traditional metaphoric strains that are thrust into the poem by the poet. However, the space available is so limited at this juncture in this rather desultory ratiocination on the optimum fusion between the literary text and

---

<sup>6</sup> The metaphoric use of the organic adjectives is far from the kind made much of by I.A. Richards when he mentions "a shock of astonishment, an astonishment which is full of delight, but none the less is a shock" (*Principles of Literary Criticism*, p. 105) in connection with Keats' *Satyr's Song*, but nonetheless effective as they are yanked together with the haggard knight in the literary context of ballad, a somewhat overused and musty literary style.

digital media. I feel compelled to hurry on and come to a conclusion, albeit temporary, that hopefully lays out a general idea to underpin the amalgamation of the sort pursued throughout this essay. If we look at the next segment of the ballad, we immediately notice that the tone that comes out of the putatively “disenchanted” knight’s narrative turns decidedly physical, to the extent that whole quatrain resonates almost with voluptuous yearning for the beauty that is unattainable. Note the first two lines.

I met a lady in the meads,  
Full beautiful, a fairy’s child;

The normative descriptive lines may beguile the average reader into believing that these lines are merely that, a descriptive statement of how the knight ran into the lady in question, and nothing more. However, the complete abandon with which the knight narrates the circumstances in which he met the enchanter attests to the strong undercurrent that runs throughout the poem, without the comprehension of which the true nature of the poem does not congeal in the reader’s mind. It may be appropriate to remind ourselves that the physical references that have been segued into at the mid point of the ballad have now surfaced fully and they are threatening to take over the tonal texture of the poem as a whole.<sup>7</sup> What has been started as a deeply introspective poem now turns into a ballad with full of corporality with added voluptuousness that almost enwraps the knight’s entity. But does the corporeal side indeed completely dominate the poem? Or is it just a way of splicing a totally discrepant strain into the ballad on the part of the poet and make the poem fully alive before the reader’s eyes with its, paradoxically enough, complexity? Most likely the latter. The more associative elements the poem carries with it the more satisfactory is the moment of

---

<sup>7</sup> At this juncture it may be apropos to note that the lines in connection with the main argument are indeed a demonstration of what Roman Ingarden calls the intentional correlates of the visual and the auditory. The rhythm and the phonetic value of the poem are so well matched with the textual significance which the written words express that without, or even simple mouthing the lines, is enough to invoke the rich texture of the knight-errant in the medieval context. The argument about the intentional correlates is so well made that I am compelled to quote the passage in full.

A normal reader who knows the phonetic form of the language well will combine silent reading with an imaginary hearing of the corresponding verbal sounds and the speech melody as well, without paying particular attention to this hearing. When the verbal sound is relatively important, the reader might even pronounce the sound involuntarily and quietly; this can be accompanied by certain motor phenomena. The auditory apprehension of the phonetic form of the words is so closely related to the visual apprehension of the written form that the intentional correlates of these experience also seem to be in especially close relation. The phonetic and visual forms of the word seem almost to be merely two aspects of the same “verbal body” (*The Cognition of the Literary Work of Art*, p. 21).

interface likely to be when the interactive reader finally comes to face to face with the core of the ballad. It is exactly in this and other non-verbal elements that the digital media embedded and threaded into the current fusion is expected to manifest their true mettle. Although textual presentations, which have been the mainstay of the printed culture of the traditional literary or other scenes in the past, it is time that the fusion adumbrated in this essay now took place and executed in a manner to bombard every nook and cranny of the interactive subject's imaginative chamber. Needless to say, the project is not going to be as simple as may have been suggested in this paper. But if there is even a gleam of attaining the light/enlightenment that has been elusive up till now, it may be a rather healthy attitude that in trying there is to be everything gained and nothing lost. By building upon the concept developed here a new level of literary appreciation is sure to be achieved and it is going to be far richer than anyone ever dreamed of.

#### Works Referenced

- Abrams, M.H. *Natural Supernaturalism*. New York: WW. Norton, 1973.  
 Abrams, M.H. *The Mirror and the Lamp*. Oxford: Oxford University Press, 1971.  
 Abrams, M.H., ed. *The Norton Anthology of English Literature*. New York: W. W. Norton, 1993.  
 Catanese, Paul. *Director's Third Dimension*. Indianapolis, Indiana: QUE, 2002.  
 Ingarden, Roman. *The Cognition of the Literary Work of Art*. Evanston, Illinois: Northwestern University Press, 1979.  
 Orgee, Mel et al. *Director 8.5: Studio*. Birmingham, UK: friends of ED, 2001.  
 Richards, I.A. *Principles of Literary Criticism*. London: Routledge, 1983.  
 Rosenzweig, Gary. *Advanced Lingo for Games*. Indianapolis, Indiana: Hayden Books, 2000.  
 Schaubert, Ellen and Ellen Spolsky. *The Bounds of Interpretation*. Stanford, CA: Stanford University Press, 1986.

#### インターアクティブな文学作品インターフェイスの構築

文学作品をデジタルメディアとの融合により、意味深長な、そして「読者」をより引き付けるような entity として提供するためには、それなりにインターフェイス上での工夫が

必要となる。この論文では文学作品の文化的側面をも包含したプラットフォーム構築の可能性を追求しつつ、非直線的、multi-valent な意味・解釈的なレイヤーの挿入を試みる。