Abstract
It has been suggested that far-eastern (predominantly Chinese and Japanese) and western reasoning styles differ greatly: Westerners focus on objects, whereas Easterners focus on fields of interaction.\[1\]

Traditional forms of object-orientation seem to follow the western style of thought, in which individual entities are captured cleanly, while interactions between them are not: It has been widely noted that many software systems that involve dynamically interacting components can be complex to design and implement using a strictly object-oriented approach.

Eastern reasoning style lends itself better to description of interactions between entities than does the Western style. Hence, we posit that programmers designing systems that involve complex interactions might benefit from a more eastern approach for their design.

However, there are currently no eastern-style programming languages of which we are aware. So, we begin our exploration with the work presented in this paper, in which we interview Easterners about how they would describe a typical object-oriented scene, and then attempt to capture and distill their descriptions into the guidelines for a programming paradigm.

Categories and Subject Descriptors D.m [Software/MISC]: Software Psychology; K.4.0 [Computing Milieux/COMPUTERS AND SOCIETY]: Social Issues; K.4.2 [Computing Milieux/COMPUTERS AND SOCIETY]: General

General Terms Design, Languages

Keywords Culturally derived programming languages

1. Introduction
In his popular book, The Geography of Thought, Nisbett [?] makes the case that individuals from different cultural backgrounds reason about objects and spaces differently from one another. Specifically, he asserts, and shows through peer reviewed studies, that subjects from Chinese and Japanese cultures do not relate in the same way to objects as those from the west. Western thinkers, says Nisbett, identify the world by objects, viewing the world as a set of individuals, working essentially independently, maintaining their own world view, and acting upon it. Eastern\[1\] thinkers, on the other hand, view the context of objects as centrally as the objects themselves; they notice changes in the scenery before they notice changes in individuals within that scenery. When considering objects, they consider the fields of interactions between those objects, rather than seeing the objects as autonomous. They look, it seems, for harmony.

It is possible that this cultural difference in reasoning may extend not only into looking at the physical world, but also into consideration of programming.

Most major programming languages, especially those in the paradigm called Object-Orientation (OO), were developed in the west, by, what Nisbett would classify, as individual thinkers. The roots of object-orientation were, of course, to help programmers model the world as they saw it; to better align their programmatic representations with their mental models of a problem space.

Problems arise, however, when systems do not align well with a pure Object-Oriented modularization. This misalignment is often evident in systems that involve a great deal of object interaction and negotiation. Almost since the rise of OO, attempts have been made to break apart the rigid adherence to the individual-nature of object-oriented languages. Aspect-orientation is a recent example of such an attempt: through aspects, developers can describe concerns that crosscut objects [?], or can capture, in one location, the relationships between them [?]. This multidimensional movement may fit into the concepts of postmodern programming, as described

\[1\] Here, we will use the term Eastern to refer to Nisbett’s grouping of the Chinese and Japanese.
by Noble and Biddle [?]. It is also possible that these attempts are edging towards capturing a more Eastern philosophy of programming, where fields of interaction are as important as the objects themselves, and where writing a working program means attaining programmatic harmony.

Inspired by Nisbett’s description of differences in Eastern versus Western styles of reasoning, and motivated by how closely the Object-Oriented paradigm resembled typical western thought, we wished to investigate the use of an Eastern reasoning approach for capturing object dynamics and interactions between objects. Since Object-Orientation grew from the minds, and reasoning style, of Westerners, we decided to look deeper into the minds of Easterners in an attempt to capture their world view, and possibly drive a new, more harmony-oriented, programming paradigm.

We undertook a small study to help us capture the way in which Easterners would describe what would be considered a typical OO scene—Nygaard’s, now famous, Restaurant Picture (shown in Figure ??). Kristen Nygaard introduced this picture as a mechanism for teaching students about OO. He motivated the use of this intricate image by saying “to teach object-orientation, you need a sufficiently complex example”. The idea, as he presented it, was that students (of any age!) would be able to look at this picture, and identify different kinds of people, their traits, and the activities in which they were engaged. This would help them think about objects; about individuals.

When Nisbett’s findings are considered in a programming context, they imply that Eastern thinkers would not interpret Nygaard’s restaurant in an object-oriented way. That, in fact, they would notice fields of interaction in the picture, and report on the harmony of the restaurant as a whole, rather than on the individuals within it. Our working assumption, based on Nisbett’s findings, was that while Westerners would likely describe the scene in a way that could be straightforwardly captured using OO concepts, Easterners’ descriptions of the scene would help identify concepts that could not be so straightforwardly captured.

Not wishing to undertake the task of re-proving Nisbett’s findings using this picture (which is similar in style to those he used in his studies), we did not set up a controlled case study comparing Eastern and Western subjects. Instead, we simply relied on his findings, and interviewed eleven Eastern subjects about how they would describe the picture. We hoped that their analysis of the picture would give us insights into how harmony might be perceived in Nygaard’s picture, and how it might be captured in a programming language.

Here, we relate the preliminary analysis of the subjects’ responses, and, based on that analysis, we suggest possible avenues for pursuing Harmony-Oriented programming: A paradigm that would allow for straightforward modeling of how objects and their behaviors affect one another. We hope that this work will give rise to a more systematic and comprehensive investigation of these ideas.

2. The Geography of Thought

[Easterners] see the big picture and they see objects in relation to their environments—so much so that it can be difficult for them to visually separate objects from their environments. Westerners focus on objects while slighting the field and they literally see fewer objects and relationships in the
Nisbett’s search for the difference between Eastern and Western thought reportedly began when his Chinese graduate student startled him with a provocative and inspiring remark. The student noted that the Chinese believe in looking at the big picture, at relationships between objects, and that they use broader perspective when reasoning about events. Westerners on the other hand, live in what the student called a “deterministic world” where Westerners erroneously believe that they “think they can control events because they know the rules that govern the behavior of objects” [7], pp xiii.

Nisbett’s student’s statement was inspirational for us as well. We had been considering how to capture fluid interactions between objects as an enhancement to aspect-orientation, and so had been heavily mired in approaches for controlling events by governing object behavior. We could very much identify with the idea of living in a deterministic world. We read the rest of Nisbett’s book with object-orientation in mind. Here is what we found.

2.1 Western Obsession with Modularization

According to Nisbett, Westerners are keenly interested in atomizing the world. He notes that this contributed largely to economic advances and industrialization: allowing manufacture to happen in generic ways, which enhanced efficiency and interoperability of approaches. He goes on to say that this modular view carries through to social infrastructure. He asked Easterners and Westerners to consider phrases that described companies as either social networks where people work together, or as an institution with a goal, where people are hired to perform functions. Most Westerners related best to the atomized view described in the second statement. Easterners predominantly chose the first statement as more accurate [7], pp 84.

Object-orientation seems to mirror the western, atomized, view of the world. A system is something that addresses some requirement (or set of requirements), and objects are formed as functioning entities that carry out tasks associated with that requirement. To shift away from the western/OO view would mean increasing the importance of the relationships between objects, and somehow capturing the field in which objects interact as a first-class concept.

2.2 Dialectic Reasoning versus Identity and Non-Contradiction

Nisbett contrasts Eastern versus Western reasoning by describing dialectic reasoning from the east, and identity and non-contradiction in the west.

Dialectic reasoning held in eastern traditions involves:

• The Principle of Change: this captures the constantly changing nature of reality.
• The Principle of Contradiction: Due to constant change, paradoxes are constantly being introduced. Both A and !A might be true at the same time.
• The Principle of Relationship: Holism: Nothing exists in isolation. Everything must be described by its relationship with other things.

Westerners, on the other hand, hold two logical principles dear:

• The Law of Identity: A is always A, regardless of context.
• The Law of Noncontradiction: A and !A cannot both hold true.

Control structures and abstractions rely on the logic described as the western style: basic concepts of truth allow programs to successfully operate. Fuzzy logic systems relax the boolean truth concept to some degree, but still do not afford the full contradiction that the second dialectic principle necessitates.

The concept of program or object state aligns well with the western Law of Identity, and flies in the face of the Principle of Change. Small irregularities in objects when operating in different contexts are seen as difficult. The classic example of non-orthogonality in programming languages is the difference in how arrays and simple variables are passed into sub-routines (the first by reference, the second by value). This contextually-imposed difference is taught in lectures, is minimized in language design, and is generally highly controlled: A must always be A.

Finally, the principle of relationship is realizable in object-orientation, but only with difficulty. Objects operate best when they have at least the illusion that they operate in isolation. Work by Noble [7] has looked at capturing relationships between objects using aspect-orientation, but this does not go as far as to claim that objects are fully described by how they relate to others. To attain such a thing in object-orientation would be analogous to attaining the buddhist concept of no self, which is quite the opposite of what much of object-orientation is about.

2.3 Control Flow

Nisbett’s research indicates that Westerners place great importance in feeling a sense of control, whereas Easterners are more likely to acknowledge that they are out of control, and make adjustments to fit into an uncontrollable situation ([7], pp 97). Adjustments for Westerners, on the other hand, were assessed to feel unnatural or “awkward”. Westerners are also more interested in knowing who is in control, as is evidenced by westerner’s dislike of working in groups, where control may be ambiguous. Easterners, on the other hand, would rather work in a group, regardless of the quality of that group, and simply adjust to the group dynamics without establishing explicit control.

Control is a basic concept programmatically: The current execution point is one of the most fundamental ideas in programming. Developers are better off during debugging tasks when they can tell what the state of the system is, and where the control is. In an OO setting, objects submit control to one another. Unless a system is explicitly parallelized, no two objects can hold the point of control at once. Neither can there be any ambiguity about which object is in control. Ambiguities, and situations where ambiguities might arise, such as access of mutually exclusive methods or data, are dealt with explicitly, in order to be ruled out as potential areas of trouble.

Relaxing the need for control in the programmatic sense would mean a fundamental shift from OO style, and away from the typical assumptions that underpin most modern programming languages. It points to the need for high degrees of interaction and collaboration between objects, but not in a synchronous or sequential manner.

2.4 The Resting State

Nisbett and his colleagues showed twelve graphs showing growth trends plotted over time:

[We] asked them to actually plot what they thought the next two data points might be. Americans were likely to continue the trend in the same direction, and at the same rate...The Chinese on average predicted a leveling off of change, and were several times more likely to predict a reversal in direction of change than Americans were

Nisbett [7]. pp 106

This led to the finding that Easterners believe in adjustment and compensation for change; that “what goes up, must come down”. He found they held a solid belief in a natural, resting state for all things, and that external disturbances in the resting state of objects are agents for impermanent change.
Dropping - resistance Floating - Collision

Figure 3. Trajectories of motion ... suggestive of liquid in a container (Nisbett, pp 117)

There is no such notion of this in traditional object-orientation. It might be the case that individual objects are programmed to return to a resting state, but this is not a fundamental principle. Instead, when an object's state is changed, it stays changed, unless that state is explicitly reverted. Adhering to this notion when considering a programming paradigm would introduce a fundamental characteristic of objects as constantly trying to return to their resting state.

2.5 Internally versus Externally Stimulated

Nisbett also found a fundamental difference in beliefs about motivations for actions or change. He showed participants the images re-drawn in Figure 3, and then performed the following experiment:

Participants were asked to what extent they thought that the object's movement seemed influenced by internal factors. The Americans reported that they thought the movements were caused by more internal factors than did the Chinese.

—Nisbett [2], pp 117

Objects can be externally motivated when they are sent a message from another object. However, objects are typically not seen as motivated by something other than an object. There is no notion in OO of gravity or some natural force affecting the movement or behavior of objects, unless that force is modeled by another object, and exerted accordingly. Objects can affect one another, but that's as far as it goes. Once again, this idea points to the notion of a first-class concept of a field in which objects operate and interact.

2.6 Substances versus Objects

Nisbett's book refers to a very interesting experiment by Imae and Gentner 2 called The Dax Experiment. In it, Imae and Gentner showed Easterners and Westerners a shape made out of some substance, and told the subject to "look at this dax". Then, they showed the subjects two trays of objects, one carrying objects made from the same substance as the dax, and other carrying objects that were the same shape as the dax. They were then asked to identify the tray with the dax on it. Westerners predominately chose the tray with objects of the same shape, whereas Easterners chose the tray with objects of the same substance.

Nisbett points out that this study indicates that while Westerners see the world as a set of disconnected objects, modern Easterners view the world as continuous masses of matter. When considering this in the context of programming, this idea motivates a shift from object-orientation to substance orientation. Objects still, to some extent, exist. But, rather than having discrete objects, objects would be made up of substances that would continuously flow into one another.

3. The Exploration

Our motivation for this work was to arrive at a set of guidelines or principles for a new kind of programming language that could capture fluid interactions between objects without heavy approaches and protracted negotiations. We wished to base this language on the Eastern style of reasoning outlined by Nisbett, as opposed to the Western style of reasoning that has motivated most programming languages to date. Kristen Nygaard said in personal communication to one of the authors that when devising Simula, he had wished to make a language that would allow direct expression of a scene very much like the restaurant picture. Inspired by his story, we followed the same process, and wished to find guidelines for a language that would allow direct expression of the eastern view of the scene in the restaurant picture.

We interviewed eleven participants about how they would describe the restaurant picture shown in Figure ??, We then examined the subjects’ responses to identify concepts that might be difficult to express in object-orientation. Finally, we examined these concepts in the context of Nisbett’s findings for broader guidance on an initial framework for an eastern-style programming paradigm.

4. Picture Description Interviews

Predictably, Nisbett’s book was missing a description of the kinds of concepts that an Eastern thinker might want to express when programming or modeling. We believed that the secret to developing a programming style that allowed harmony was to note the kinds of observations that would be made by Easterners to describe a scene that would otherwise be modeled in OO. The ultimate goal of this was to derive the framework for a language: the kinds of statements that should be expressible to capture harmony.

We conducted eleven interviews with individuals from China (a mix of those from Mainland China and Hong Kong). In the sessions, we simply asked the subjects to describe the scene shown in Nygaard’s picture. We instructed them to describe anything they saw, with no restriction. We let them speak for as long as they felt comfortable (durations ranged from 2-8 minutes). We also allowed them to annotate a copy of the figure if they wished. We suggested that this might help them clarify the portion of the picture to which they referred. We did not inform them that they were being interviewed because of their cultural background. Instead, we indicated that we were conducting a study on how pictures are described by different individuals.

We performed two passes over the transcripts of the interviews. We initially performed a cursory read to derive general commonalities between the descriptions, and arrived at a set of statement-categories. We then performed a detailed analysis to categorize each statement made by each subject. As Nisbett’s findings predicted, the subjects did not comment to any great extent on individuals or their traits. Instead, any descriptions of individuals were couched in descriptions of how an individual related to a group. We identified three categories into which the subjects’ descriptions of the scene could be grouped: Context, Relationships and Puzzlement. We further decomposed the first two of these into the following categories:

- Context: Environment
- Context: Interpretation
- Context: Observation
- Context: Role
- Relationship: Short Range

---

• Relationship: Long Range

Remarks could be categorized under multiple categories.

4.1 Descriptions of Context

Context refers to a description of the situation in which objects are placed. As mentioned above, we sub-categorized this into:

• Context: Environment
• Context: Interpretation
• Context: Observation
• Context: Role

4.1.1 Context: Environment

This category covers remarks that pertain to the environment of the restaurant. For instance, one subject stated:

And I think this restaurant does not have too much lights, just a few. And I just wonder where ... maybe the environment is very dark

—Respondent 2

Other remarks placed objects (people in the restaurant) in context:

Oh, and actually there is a piano in the ... at the corner of the restaurant. It is in a dark place, so I cannot notice it until now. And seems the man is playing to himself only. Since he is sitting back to the people, so he enjoy, oh no, it seems he is not very enjoy in his piano playing. 'Cause I see his mouth seems not smiling, and it is very dark. Seems he is working alone...

—Respondent 6

We found seven quotations that fit into this category.

4.1.2 Context: Interpretation

This category covers remarks where the respondent is interpreting what is going on in the restaurant, and perhaps trying to determine the reality behind something they perceived to be unclear. For instance:

...this one...maybe just arrived the restaurant and then want to put the clothes here. Or the opposite: Finish eating and then take the clothes from her.

—Respondent 7

Also:

And a kid is crying, but her mom seems not to take care of him

—Respondent 2

20 remarks were categorized as Context: Interpretation.

4.1.3 Context: Observation

This category includes remarks that are simply observations about the restaurant in general (not the environment or ambiance). For instance:

I think most people here are not eating at all. They are having drinks or chitchatting

—Respondent 2

We identified 21 remarks that fit into this category.

4.1.4 Context: Role

In this category, we simply identified the different roles that respondents identified. Respondents identified:

• Waiter (Respondents 1, 4, 6, 7, 8, 11)
• Pianist or Musician (Respondent 3, 8)
• Cook or Chef (Respondent 3, 6, 7)
• Boss (Respondent 4)
• Doctor (Respondent 4)
• Workers (Respondent 4)
• Guest (Respondent 8)

We identified 17 remarks related to roles.

4.2 Puzzlement

We identified 15 remarks that relayed confusion about the restaurant scene.

For instance, Respondent 9 remarked that the piano area of the restaurant is cut off from the rest of the restaurant:

And I think this corner is quite impossible. How come they can just go there? Because ... the details ... is not shown in this picture

—Respondent 9

Another respondent commented that the man at the coat-check did not fit in well with the rest of the guests:

And in the counter, there is a people who is wearing ... too much because there are, for other people, they are just wearing as normal summer, but this one like coming from the north pole

—Respondent 1

4.3 Relationships

We identified two categories related to relationships between people in the restaurant: long and short distance. Short distance relationships are between people at the same table (or in some way involved with the table), whereas long distance relationships are between tables.

Figure 4. Long-Distance Relationships Identified
4.3.1 Long-Distance

Long distance relationships refer to remarks that describe how two sets of people relate, when those people are not seated at, or in some way involved with, the same table.

We identified 10 such quotations, some of which were duplicates of others. They are depicted in Figure ?? as white lines linking the groups of people included in the remark. The longest distance relationship spanned from the highlighted area A to the highlighted area B. A shows two people from table 20, and B highlights what is identified as a cake. The two respondents who noted this relationship stated that the people in area A were watching and waiting for their cake, which was being brought to them from area B.

4.3.2 Short-Distance

As described above, short distance relationships refer to remarks describing how people at a single table relate. We found the following short distance relationships:

- Mother and Son (table 1—Respondent 2, 4, 6, 7, 11) (table 7—Respondent 3)
- Friends (Respondent 3, 4, 7)
- Dating (Respondent 3, 4, 11)
- Family (Respondent 6)

4.4 Common Observations

Several observations were made by multiple respondents. Table ?? (in Appendix A) provides the results of the analysis of the observations, and shows which category the responses fit under. Some observations fit into multiple categories. The total number of responses counts the individual responses from all categories; the same respondent may have uttered more than one response for a particular observation. The number of individual respondents who reported an observation is shown in parentheses under Total Number.

The rest of this section delves into more detail about the four most popular observations.

4.4.1 Table 6: Man Unable to Pay

Three respondents commented on the man at Table 6 (highlighted in area “A” in Figure ??).

Figure 5. (A) Man at Table 6 Unable to Pay; (B) Women at Table 4 perceived to observe him

It seems they want to pay; the man is searching for purse; maybe he loses it. The waiter is waiting for him

—Respondent 3 (Context: Interpretation)

4.4.2 Empty Seats, but People are Waiting

Several of the participants in the study noted, with consternation, that there were people waiting outside the restaurant while there were empty seats inside (at Table 3). They wondered why the waiter would not seat people at that table, and commented that the restaurant was “strange” because of it.

For instance, one participant exclaimed:

But ... there is some people waiting outside. I wonder why ... why the waiters doesn’t let them sit on these empty seat

—Respondent 2:6

And I think this table is quite odd. This is not occupied. But someone are waiting outside. To waiting to have their dinner in this restaurant.

—Respondent 9:4

4.4.3 The Bad Mother

Another great cause for concern among the participants was the mother sitting at Table 1 with her child (shown in Figure ??). Five of the participants noted something very similar to one another: that the mother is covering her ears attempting not to hear her child crying:

The son is crying, but the mother covers her ears and pretends not to hear

—Respondent 3:8

4.4.4 The Misplaced Chef

The fourth most frequently observed element of the restaurant was the chef, who is, apparently, out of place. Many of the respondents...
Figure 7. The Mother is perceived to ignore her crying child

noted that the chef was not in the kitchen, that he was lazy, or that
he was simply taking a break.

How come there’s a cook here? Very strange, don’t know
why
—Respondent 3:15

Figure 8. The Chef is perceived as out of place; to others, he seems
to be crying

5. Analysis

After performing a qualitative analysis of the responses, we noted
that all of the most common responses (listed in Table ??) could
be distilled into two major kinds of observations: Identification of
disharmony in the restaurant, and collective action.

5.1 Identification of Disharmony

The respondents were overwhelmingly troubled by the “strange” or
crazy nature of the restaurant, where things were not as they should
be, and where imbalance reigned supreme. This generally fit into
different categories:

- People or things being physically out of place
- People who did not fit in, visually, with others
- People shirking their responsibilities
- Things simply going wrong

5.1.1 People or Things Physically Out of Place

There were many instances when respondents noted that people or
things were not where they should be, but the two most common
were the out of place cook and the non-sequential table numbers.

The respondents who commented on the cook all noted that he
was behaving strangely, and that his place was in the kitchen. They
wondered why he was not there now! This can be interpreted as an
identification of disharmony, because the cook’s resting state was
to be working in the kitchen, and he was upsetting the balance of
the restaurant by leaving that state and emerging into the dining
area.

The respondents who commented on the table numbers
seemed jarred by what they perceived to be a gap in the contin-
uous space of the restaurant. This was also found in the statement
about how a guest might climb up to the piano area (a frustrated
respondent noted that there was no clear way to get up to there). The
respondents suggested solutions to the disharmony identified:
there must be more tables that aren’t shown in the restaurant, and
there must be hidden stairs somewhere.

Another example of a discontinuity was the perceived lack of
teenagers in the restaurant. The respondent who noted this iden-
tified all other age ranges, but said “why are there no teenagers
here?”. No solution was given for this discontinuity.

5.1.2 People who do not “fit in” visually with others

Many respondents referred to people who did not look quite right,
and described this visual disharmony as certainly the fault of the
person who is improperly dressed. Some respondents tried to find
reasons that the people were crazily dressed, not willing to simply
accept that it might just be a scene that was beyond reasonable
explanation.

For instance, the man by the coat check, who is comedically
dressed in north pole regalia, and who is holding a fish on a skewer,
was the source of much concern for several respondents. Explan-
ations for why he looked the way he did was that he had found
his way into the wrong location, thinking he was going to a bar-
becue, and that he was having take out (the fish, we suppose). The
respondents noted that he might be about to take off his coat (hence
adjusting himself to blend in with the other diners).

The other popular observation about visual disharmony was the
strange table of party goers. This group wore costumes, and it was
remarked upon by three respondents that it is not clear “what kind
of people” sat at this table, because not only did their clothes not fit
in with the restaurant, they were not even a cohesive group.

“There is very strange, because I see that two people on the
table is like going to a party. But then other two of them,
some kind of professional people, because one look like a
doctor or dentist, and another one is something related to
some religions”
—Respondent 1:2

5.1.3 People who are not fulfilling their roles

Respondents were quick to point out that some people in the restaur-
antly were stubbornly disturbing the peace. The mother with her
crying child refused to perform her duty and calm her child. The
waiter was being careless, and was about to drop all the plates,
which would further disrupt the harmony of the restaurant. It was
perceived that none of the guests in the restaurant were doing their
jobs all that well, because no one was eating or having a good time!
Some respondents noted that guests seemed upset or distracted in
some way. These observations echo Nisbett’s findings that Eastern-
ers see a strong relationship between roles and harmony.

5.1.4 Other Imbalances

Respondents noticed other situations in which harmony was not
maintained. The most common were the imbalance between the
empty tables and the people waiting for tables, and the lack of light
in the restaurant. The number of non-working lights was remarked
upon by more than one participant, and the concept of lighting or darkness was noted by several. Since running this study, we have noticed that Chinese restaurants are incredibly bright, and that everyone is more or less continuously eating. There is always food on the table, even when the meal is clearly finished, and if drinks are being had. In a sense, it seemed that the restaurant itself was perceived to be not fulfilling its role: of being a bright place where people are happy and eating.

5.2 Collective Action

Other responses could be categorized as “actions”. However, unless they were counted as disharmonious (the mother with her child, the strange man with the fish, the bad waiter), they were described as collective action: a group of people doing something together, and feeding off each other in doing so. Once again, the party-going table figures in: they are perceived to be actively waiting for their cake to arrive (both anticipating, and looking at the arriving cake). Another popular example of collective action was the three women sitting together and laughing at the man who could not pay his bill. Finally, respondents identified a group at a table who were all trying to gain the attention of the pianist by waving to him.

There was no sense that the actors in the groups were individuals working together - instead, the actors seemed to gel in the words of the respondent into one larger actor: the group. The individuals were at times described as behaving specifically within the group, but the action belonged completely to the group, not to one of the people in it.

6. Considerations for Harmony-Oriented Programming

After analyzing the commonalities between subjects’ responses we found that they were further categorizable as descriptions of harmonious action, and of harmonious situations and flow.

Both of these concepts play important roles in current software systems, even if they are not easily attainable. For example, in a load balancing system, a harmonious situation, or equilibrium would be achieved when various instances of a component share the same workload. This equilibrium is lost if some instances are busy while other instances are idle.

Similarly, harmonious situations are the goal for system monitoring and management software, such as intrusion detection, virus detection, and network monitoring systems. The purpose of such systems is to preserve the harmony of a system of components. Such components are in harmony if they operate at full capacity and are able to interact with each other without any disturbance. The balance of this harmony can be disrupted by system anomalies, which might be caused by viruses, intruders, or component failures.

Harmonious action and situation play a significant role for the components of a single software application or server. Within an application, a harmonious situation is achieved if all components are working well. However, each time a component of the system is updated or otherwise changed, the balance of the system is disturbed and has to be restored. This is especially the case when component interfaces are adjusted or components with critical bugs are added to the application.

On a finer grain, harmonious action between objects in a system might be found when they are communicating well, and using interfaces provided by one another correctly. In traditional systems, attaining this harmony involves negotiation between objects; Ensuring up front that such harmony will be achieved involves checks by the compiler.

We now examine the concepts of harmonious situation and harmonious action from the perspective of modeling these concepts.

6.1 Harmonious Action

The observations related to collective action were all described in terms of two sets of entities: a group of actors, responding to some stimulus, or attempting to communicate with another actor:

- Women laughing at the man who cannot pay: Actors: women; Stimulus: Man unable to pay
- A group of people waving to the pianist: Actors: the people; Other actor: the piano player
- The table of people looking at and waiting for their cake: Actors: the party table guests; Stimulus: the cake arriving.

In all of these situations, the respondents seemed to be describing a situation in which entities were sitting on a surface, where action by one entity would produce ripples in the surface which would stimulate a response. For instance, in the upper portion of Figure ??, two objects are shown. One is actually a group of other objects, and one is the stimulus object. The group of objects sit in their own harmony field, where the entities form a completely connected graph.

In a typical OO scenario, these two objects would need to be connected by something fairly heavy and infrastructural. Precisely the type of stimulus would need to be known to connect the objects. However, the language used to describe entity interaction by the respondents seems to suggest that heavy-weight and situation-specific connections between objects will not suffice. This supports what Nisbett has found about negotiation-style in Etaters: He notes that “disputants take their case to a middleman whose goal is not fairness but animosity reduction — by seeking a Middle Way through the claims of the opponents” ([?], pp75).

In the harmonious action described by the respondents, the stimulus entity disrupts the harmony field, and causes a response. The group of entities respond to both the disruption of their grouping, but also to the disruptions of their neighbors. This is depicted in the lower portion of Figure ??.

![Harmonious Resting State](image1)

**Harmonious Resting State**

- Group of entities
- Stimulus entity

![Harmony Disturbed; Disturbance Propagated](image2)

**Harmony Disturbed; Disturbance Propagated**

- Group of entities
- Stimulus entity

Figure 9. Harmonious Resting State: The resting state of a harmonious system. Objects are connected in a harmony field (shown as a wire). Harmony Disturbed: The harmony of the field is disturbed by the stimulus object. That disturbance is propagated to the group of entities, and is further felt within the group.
6.2 Harmonious Situations and Flow

Many of the observations (especially those under the category of Context:Observation) took the form of an observation that identified disharmony, which in turn, implied what constraint would need to be satisfied for harmony to be attained. For instance, a number of respondents noted that there were empty tables, but that there were also people waiting. This observation identified disharmony. The implication of that observation is that people should be seated at the table (perhaps by the waiter).

This concept of harmony is discussed by Nisbett, when he discusses the Japanese awase, meaning harmonious, or fitting in. He notes that awase “rejects the idea that man can manipulate the environment, and assumes instead that he adjusts himself to it” ([7], pp76).

For instance, in the case of the empty table and people waiting (ignoring the waiter for now), the table emptying would cause a topology change in the field. Consequently, the waiting group would be tilted towards the empty table entity, and there would be a flow in the field from the waiting group to the table entity. This is depicted in the abstract in Figure ??.

Figure 10. Self-Adjustment to a New Situation. Since the entity is connected to the field in which it sits, a change in that field results in a self-adjustment. This figure shows an entity with some internal substance, and two conduits through which that substance can flow. During the resting situation, nothing is flowing from the entity. When the field topology is changed, the entity tips with the field, and its substance flows into the field.

Resting State

Flow Stimulated

7. Principles for Harmony-Orientation

We now extrapolate from the analysis of our study to identify the core concepts that would go into an eastern-style harmony-oriented programming approach. Based on our analysis, we believe that the approach should consist of two major portions: the field, and the object. Both of these are as important as one another. The field stimulates behavior by the objects, but, reflectively, responds to the movement of the objects. We provide a simple sketch of how the field and the objects might be described in such a paradigm, and then use those descriptions for a simple application: pipe and filter.

7.1 The Substance of the Field

Objects should all be placed in a field that they are all, in fact, “one with” (that is to say that they are essentially made up of the same substance as the field, and that the substance of the field flows into and out of them). The field should be described separately from the objects to capture, essentially, its physics: how quickly disturbances are propagated, how quickly substances flow, and how quickly that flow is dampened; how “heavy” an object must be to change the topology of the field, and whether the field is elastic in its response to a change in topology.

The field would allow for external stimulation of objects. As was suggested in the earlier section, this could be modeled by objects acting as containers for the substance of the field, and also sitting in the field and being acted upon.

In the following example, the field is depicted as a line, but it could just as easily be a plane, or even be multidimensional. The shape of the field should be describable, and perhaps even changeable dynamically as the system runs.

7.2 The Substance of an Object

Based on both Nisbett’s findings, and our own analysis of respondents describing the Restaurant scene, objects in an eastern programming model should be described by:

- their resting state
- their relationship to other objects (their place in the field)
- their roles and responsibilities
- the direction in which they can instigate flow (essentially, their range of influence)

7.3 A Simple Use of Harmony: Pipe and Filter

Figure ?? shows a very simple application of harmony modeling. There are three kinds of objects in a pipe and filter system: the source, the sink, and the filters that sit between them. We chose this example because it captures a very simple instance of objects working together to accomplish some goal.

Here, we show how pipe and filter might be modeled in a harmony-oriented system, and work through the three typical application scenarios: push, pull and mixed. First, we describe how the harmony field would be set up for pipe and filter. Then, we describe how the objects themselves would work within the field.

7.3.1 The Field

In this case, our field has strange physics: The field is elastic, and is re-shaped when an entity moves vertically. If an object moves down, the field will eventually spring it up a corresponding amount before returning it to its resting state.

Flow moves downwards in a reshaped field.


7.3.2 The Objects
All the objects are shown to have a resting state: that is, sitting on the horizontal line of the field. Objects move in reaction to changes in the topology of the field. As they move, they push up and down the topology of the field, and instigate flow.

The source, sink and filters are all placed evenly in a line in the field.

The role of the objects is to accept inward flow, and to propagate outward flow (if they do so). The role of the source is just to propagate flow, but not to take any flow in. The role of the sink is to take in flow. The role of the filters are to do both. The objects grow as flow is taken on, and shrink once they are empty of substance.

Dashed arrows along the field indicate the direction in which flow can be instigated. The entities can only send data to the right; the sink entity does not instigate any flow. Objects are stimulated to move vertically by being pushed upward in the field by a disturbance, or by being filled with data. Revisiting the action of the field: if an object falls in the field, it will be filled with data, and then be rebounded upwards a corresponding amount by the elasticity of the field.

7.3.3 Illustration of Dynamics
There are two typical approaches for implementing pipe and filter systems: push and pull. In a push model, the source pushes data at the filters, which in turn push data at the sink. In a pull model, sink actively obtains data from its preceding filter, and in turn the filters actively obtain data from the source. A mixed approach allows filters to instigate data flow by both pushing and pulling data.

In an Object-oriented system, functionality enabling the pull, push and mixed models must be added separately to each object. In the harmony-oriented system, the models are simply achieved by different initial movements in the field.

Here, we illustrate the push and pull models in a harmony field. In the push model, the source instigates flow by rising due to some external stimulation (1). This alters the topology of the field, and propagates flow downward to the adjacent filter (2). That filter then falls (3) and rebounds (4) which causes it to propagate flow onward (5). This continues until the sink is reached: the sink falls and rebounds, but does not move flow onward.

In the mixed model, the first filter falls (1), instigating flow from the source (2). It then rebounds (3) and pushes data toward the subsequent filter (4), which in turn causes that filter to fall (5), rebound (6) and push data onward (7). Finally, the sink receives the data (8), and rebounds (9). In the pull model, the sink falls (1), but since it is empty it does not rebound. Instead, it drags down its adjacent filter (2), which drags down its adjacent filter (3). This prompts flow from the source (4), which causes the left-most filter to rebound (5) and push flow to the right-most filter (6). That filter then rebounds (7), pushes flow onward (8) to the sink, which then rebounds and rests (9).

7.4 Implementing the Field and the Objects
The focus of this work is not to arrive at a working solution for the harmony-oriented paradigm. However, here we briefly mention how such an implementation could take shape.

There are a myriad of ways that a harmony-oriented paradigm could be implemented. Perhaps the simplest is to build a simulation-style machine using an existing programming language. That simulation would be the field. The field’s description could be augmented using a specialized programming language to describe its physics and dimensions.

Objects placed within the field could be described using a specialized programming language that ensures that all portions of an object are described. The field component would be able to affect behavior of the objects based on their positions, roles and responsibilities.

8. Conclusions
In this work, we explored the idea of developing an eastern model of programming based on the concept of harmony. We based our work on ideas from The Geography of Thought by Nisbett [7], and also on the responses we obtained from Easterners about how they would describe a typically object-oriented scene. Our goal was not to try to arrive at a programming paradigm that would compensate for any shortcoming in Easterners when considering objects. Instead, we wished to exploit their apparent enhanced ability to identify and describe interactions between objects to better capture dynamic and highly interconnected systems.

We distilled both the responses and the eastern thought concepts into a sketch of an approach for incorporating the eastern idea of harmony and object-relatedness into a programming paradigm. This meant incorporating two concepts: the idea of a field of interaction between objects, where objects interact more with the field than with one another, and the specification of simple objects that are not autonomous, but are motivated into specific kinds of action based on the state of the surrounding field.

We will leave the exploration of a concrete implementation to future work.
<table>
<thead>
<tr>
<th>Remark</th>
<th>Total</th>
<th>Context</th>
<th>Context:</th>
<th>Context:</th>
<th>Rel:</th>
<th>Rel:</th>
<th>Puzzlem’t</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Env</td>
<td>Int</td>
<td>Obs</td>
<td>Role</td>
<td></td>
</tr>
<tr>
<td>Empty Seats, but People are Waiting</td>
<td>6 (5)</td>
<td>2:1</td>
<td>1:5, 2:1</td>
<td>2:6, 6:5</td>
<td>8:8, 11:11</td>
<td></td>
<td>2:6</td>
</tr>
<tr>
<td>Man unable to pay</td>
<td>6 (4)</td>
<td>2:8, 3:3</td>
<td>3:10, 4:3</td>
<td>3:1, 7:3</td>
<td></td>
<td>2:6, 3:8,</td>
<td></td>
</tr>
<tr>
<td>Misplaced Cook</td>
<td>6 (4)</td>
<td>3:10, 4:3</td>
<td>3:1, 7:3</td>
<td></td>
<td>6:3, 7:12,</td>
<td>11:4</td>
<td></td>
</tr>
<tr>
<td>Looking at the Piano Player</td>
<td>4 (3)</td>
<td>3:10, 4:3</td>
<td>3:1, 7:3</td>
<td></td>
<td>6:3, 7:12,</td>
<td>11:4</td>
<td>3:11, 4:10</td>
</tr>
<tr>
<td>Tables Unaccounted For</td>
<td>3 (3)</td>
<td>2:2, 6:6,</td>
<td>8:12</td>
<td></td>
<td>2:6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strange Table</td>
<td>3 (3)</td>
<td>2:2, 6:6,</td>
<td>8:12</td>
<td></td>
<td>2:6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Man “From the North Pole”</td>
<td>3 (3)</td>
<td>6:12, 7:11</td>
<td></td>
<td></td>
<td>6:12, 7:11</td>
<td></td>
<td>1:4</td>
</tr>
<tr>
<td>People are Drinking, Not Eating</td>
<td>3 (3)</td>
<td>2:4, 8:1,</td>
<td>9:3</td>
<td></td>
<td>2:4, 8:1,</td>
<td>9:3</td>
<td></td>
</tr>
<tr>
<td>Busy Workers</td>
<td>3 (2)</td>
<td>4:6</td>
<td>8:6</td>
<td></td>
<td>4:6</td>
<td>4:18</td>
<td></td>
</tr>
<tr>
<td>Plates will Fall</td>
<td>2 (2)</td>
<td>6:4, 11:1</td>
<td></td>
<td></td>
<td>11:1</td>
<td>11:1</td>
<td>2:7, 4:13</td>
</tr>
<tr>
<td>Crying Chef</td>
<td>2 (2)</td>
<td>1:3</td>
<td></td>
<td></td>
<td>8:10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strange Picture</td>
<td>2 (2)</td>
<td>1:3</td>
<td></td>
<td></td>
<td>8:10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dark Restaurant</td>
<td>2 (2)</td>
<td>6:7</td>
<td></td>
<td></td>
<td>6:7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sad Pianist</td>
<td>2 (2)</td>
<td>6:8</td>
<td>6:8, 11:2</td>
<td></td>
<td>6:8, 11:2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>People Outside</td>
<td>2 (2)</td>
<td>6:13</td>
<td></td>
<td></td>
<td>8:2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Man Under Table</td>
<td>2 (2)</td>
<td>4:14, 7:9</td>
<td></td>
<td></td>
<td>4:14, 7:9</td>
<td></td>
<td>7:8</td>
</tr>
<tr>
<td>Cake Coming</td>
<td>2 (2)</td>
<td>3:14, 4:15</td>
<td></td>
<td></td>
<td>3:14, 4:15</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>