Tagmemics

An Introduction to Linguistics for Perl Developers

or

"Wouldn't know a tagmeme if it bit me on the parse."

Allison Randal
University of Portland
YAPC::EU 2002
In the Beginning...

Wants pawn term dare worsted ladle gull hoe lift wetter murder inner ladle cordage honor itch offer lodge, dock, florist. Disk ladle gull orphan worry putty ladle rat cluck wetter ladle rat hut, an fur disk raisin pimple colder Ladle Rat Rotten Hut...

- Howard L. Chace, *Anguish Languish*
Why Linguistics?

• Linguistics and Software Development?

• Linguistics is the study of language.

• Programming languages are human languages.

• It's the same brain.
Why Tagmemics?

- Out of hundreds of theories, why this one?
- Understand Larry.
- Know your roots.
- Tagmemics is a practical theory.
Who's Pike?

- Dr. Kenneth L. Pike, 1912 - 2000
- Professor of Linguistics at the University of Michigan.
- President of the Linguistic Society of America and SIL International.
Who's Pike?

• Nominated for the Nobel Peace Prize on 15 consecutive years.

• Also known for his entertaining stories, poetry and songs and his blue cape.

• He would fit right in the Perl community.
• Pike developed tagmemics while doing field research and teaching linguistics.

• He wanted a theory that was easy to learn and easy to use...

• ...but complex enough to explain real language.

• The Swiss-Army knife of linguistic theories.
Language in Context

• Language cannot be separated from the human context in which it is used.

• Many linguistic theories are satisfied with structure alone.

• Those theories have no explanation for:

  *Wants pawn term dare worsted ladle gull*...
Any element can be analyzed in one of 3 ways:

- **Particle**: each element is a discrete unit.
- **Wave**: elements are defined by a nucleus.
- **Field**: relationships between elements are what matters.
Particle, Wave and Field

• The 3 analyses applied to "versions of Perl":
  – Particle: each version stands on its own.
  – Wave: a release is a nucleus, but development is continuous.
  – Field: versions contrast by their sets of features.
What's a Tagmeme?

- A tagmeme is a unit-in-context.
- Tagmemes are fractal.
- A tagmeme has 4 parts: slot, role, class and cohesion.
- Don't get too tied up in the details...
What's a Tagmeme?

<table>
<thead>
<tr>
<th>Slot</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role</td>
<td>Cohesion</td>
</tr>
</tbody>
</table>

© 2002 Allison Randal
What's a Tagmeme?

Where does the unit go?
Why is this unit here?
What's its function?

What kind of unit is it?
How does this unit relate to other units and to the wider context?

Role
Class
Slot
Cohesion
The monkey bit me.
What's a Tagmeme?

<table>
<thead>
<tr>
<th>Margin</th>
<th>Pronoun</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject</td>
<td></td>
</tr>
<tr>
<td>Actor</td>
<td>Obligatory</td>
</tr>
<tr>
<td></td>
<td>Number agreement with the verb</td>
</tr>
<tr>
<td></td>
<td>Case marking</td>
</tr>
</tbody>
</table>

He bit me.
What's a Tagmeme?

Nucleus
Predicate

Verb

Statement

Transitive
Number agreement with the subject
Case marking

The monkey bit me.
The monkey bit me.
What's a Tagmeme?

Nucleus
Predicate

Verb (function)

Imperative
Obligatory

print $animal;
What's a Tagmeme?

Role

Class

Slot

Noun (variable)

Optional

Object

Margin

Undergoer

print $animal;
What's a Tagmeme?

Margin
Object

Undergoer

Noun (string)
Optional

print "antelope";
What's a Tagmeme?

<table>
<thead>
<tr>
<th>Margin</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Undergoer</td>
<td>Optional</td>
</tr>
</tbody>
</table>

print join("\t", @animals);
Etic and Emic

- Two different perspectives on the same data.
- Etic is logical, alien, external, invented scientifically measurable.
- Emic is relative, native, internal, discovered, what matters to participants.
Etic and Emic

- Etic is E.T.-ic.
- Emic is ME-ic.
An etic view of a television:
- vaguely cubical, but irregular shape
- dark color
- dimensions range from inches to yards
- one surface lighted with changing colors
- emits sound
Etic and Emic

• An emic view of a television:
  – the colors represent human activity
  – a connection to the world
  – or an escape
  – an education tool
• An etic perspective on language design:
  – X number of control structures...
  – X number of forms to express them.
An emic perspective on language design:
- Who is going to use it and how?
- What other concepts are they going to be familiar with?
- What will they learn easily?
- What will be meaningful?
- How will it relate to other aspects of the language?
switch (c) {
    case '0':
        zero++;
        break;

    case 'a':
        alpha++;
        break;

    default:
        nomatch++;
        break;
}

© 2002 Allison Randal
```c
switch (c) {
    case '0':
        zero++;
        break;

    case 'a':
        alpha++;
        break;

    default:
        nomatch++;
        break;
}
```
switch (c) {
    case '0':
        zero++;
        break;
    case 'a':
        alpha++;
        break;
    default:
        nomatch++;
        break;
}

© 2002 Allison Randal
given (c) { 
  case '0':
    zero++;
    break;

  case 'a':
    alpha++;
    break;

  default:
    nomatch++;
    break;

}
given (c) {
    when '0':
        zero++;
        break;
    when 'a':
        alpha++;
        break;
    default:
        nomatch++;
        break;
}
given (c) {
    when '0':
        zero++;
        break;
    when 'a':
        alpha++;
        break;
    default:
        nomatch++;
        break;
}
given (c) {
  when '0' {
    zero++;  
    break;
  }
  when 'a' {
    alpha++;  
    break;
  }
  default {
    nomatch++;  
    break;
  }
}
Re-analysis

given (c) {
    when '0' {
        zero++;
        break;
    }
    when 'a' {
        alpha++; 
        break;
    }
    default {
        nomatch++; 
        break;
    }
}
given (c) {
    when /glick/ {
        zero++; break;
    }
    when 'a' {
        alpha++; break;
    }
    default {
        nomatch++; break;
    }
}
given (c) {
    when /glick/ {
        zero++; 
        break;
    }
    when Acme::Snark {
        alpha++; 
        break;
    }
    default {
        nomatch++; 
        break;
    }
}
given (c) {
    when /glick/ {
        zero++; break;
    }
    when Acme::Snark {
        alpha++; break;
    }
    default {
        nomatch++; break;
    }
}
Re-analysis

given (c) {
    when /glick/ { 
        zero++; 
    }
    when Acme::Snark { 
        alpha++; 
    }
    default { 
        nomatch++; 
    }
}
Re-analysis

given (c) {
  when /glick/ {
    zero++;}
  when Acme::Snark {
    alpha++;}
  default {
    nomatch++;}
}

© 2002 Allison Randal
given ($c) { 
  when /glick/ { 
    $zero++; 
  }
  when Acme::Snark {
    $alpha++; 
  }
  default {
    $nomatch++; 
  }
}
Designing a language without considering that people are going to use it is about like designing a car without considering that people are going to drive it.
Further Reading


(Both titles are out of print, but they are available at several used book sellers online.)