Consideration of naming mechanisms requires considering three separate functions: inferring intent, identifying a resource, and locating that resource (or alternatively, providing information about how that resource is served).

The trouble with naming using URIs to identify resources is that they are an indirection to an indirection. They generally require name resolution, which may vary based on location or context, and may further return multiple addresses (or address families), which may be indistinctly identifiable as an “endpoint.”

Further, as the CFP indicated, input methods may generate varied versions of users’ attempts at a URI. Voice recognition systems are notoriously bad at handling accented variations of a language, homophones, and speech pathologies. Facial recognition systems are notoriously bad at reading the variety of possible colors of faces and culturally-specific expressions. These are, in effect, the same problem as the Unicode collisions (or divergences), where the identical character or string or phoneme can mean different things in different languages, or multiple characters or strings can be used to indicate the same thing.

This may simply be a technology lag, which will be resolved in time, possibly with local training (i.e., each user teaches the voice recognition system how they pronounce words). Local training, however, will only exacerbate the problem of local context being significant.

This may disenfranchise users for whom English is not their first language, or who have a language pathology.

The naming problem, therefore, can be decomposed into the series of problems:

1. Inferring intent is local. While users can be taught limited universal communication strings, Unicode allows variation even in apparently identical strings, and the correlation between a word and its phonemes is many-to-many.
2. Identifying the resource desired varies based on context. Split DNS, local resolution, geolocated responses, and content filtering may all return different responses. Often, location information is inferred in a query or implied in a response.
3. The target of communication may still respond with resources based on its own intent.

The only solutions I can conceive for the inference of intended string are:

1. Development of a user-specific database of languages and eccentricities. Such a database would be taught the user’s peculiar dialects, whether oral, typed, gestural, or other. In order to be useful, global systems would have to identify the user and access their unique database in order to interpret their input.
2. A generalization of that, where a user could specify their default languages and dialects and eccentricities. For instance, a user might be able to specify oral languages English:American:Texas:Stutter and Spanish:Catalan:Lisp, in a search order, so that a
system would search for phonemes using the first interpretation first, (unless context such as sentence resets the search context). User-specific eccentricities might still be required, such as a Norwegian speaker who composes a Turkish character as they were raised, not as specified in Unicode, or gestural variants for people missing appendages.

To some extent, this is a service already provided by search engines. Many, however, use locale to assume a language context, an experience sometimes frustrating to very mobile users.

The ambiguity of which resource is identified is a matter of policy of the resource owner, or less often, the network owner. A resource owner may define a policy of directing to the nearest instance of a resource (using variable DNS responses based on geolocation, or anycast), or a walled garden requiring authentication. A network owner (such as a corporate network) may redirect to a page describing acceptable use. A political body may redirect (or require the network or resolver to redirect) to locally-approved versions, or Error 451. A mechanism for making such policy explicit might be useful for users and researchers, especially for debugging; the SUPA Working Group’s event-condition-action policy model abstraction might work. Locator-identifier splits are not generally necessary, but it would be nice if they could be explicit.

Even when reaching the ostensible repository of a resource, the target of communication may provide varied responses. For instance, based on cookies or IP address or other heuristics, a server might still redirect to another server, or return custom content. This again is generally in response to the content holder’s local policy, with the same considerations as in resource identification. The idea of a “target of communication” is vague, and the definition of an endpoint is a slippery one, still being discussed.

Areas for work to improve people’s ability to identify and reach resources, therefore should include:

- Research into inferring intent,
- Guidance on creating identifier systems, including articulation of principles of uniqueness, parameter overloading, collisions, locators, specificity, and how identifiers change.
- Specification of policy language to make explicit which possible response to a query will be sent.